CZERWIEC 1986 CENA 100 ZŁ
ISSN 0860-25-14

KOMPUTER 3

POPULARNY MIESIĘCZNIK INFORMATYCZNY



KOMPUTER NR 3

Popularny Miesięcznik Informatyczny – pismo miłośników i użytkowników mikrokomputerów redagują:

Marek Młynarski (red. nacz.)
Władysław Majewski (z-ca red. nacz.)
Grzegorz Eider (sekr. red.)
Elżbieta Bobrowska (z-ca sekr. red.)
Grzegorz Czapkiewicz (programy)
Darosław J. Toruń (gry)
Tomasz Zieliński (listy)
Krzysztof Krupa
oraz współpracownicy:
Andrzej Bączyński (Łódź),
Rafał Brzeski, Marek Car, Andrzej Kadlof, Jarosław Kania, Agnieszka i Zbigniew Kasprzyccy, Krzysztof Kuryłowicz (Łódź), Jacek A. Likowski, Juliusz Rawicz, Leszek i Zenon Rudak, Grzegorz Szewczyk, Jakub Tatarkiewicz, Piotr Norbert Tymochowicz, Roland Wacławek (Katowice), Tadeusz Wilczek, Andrzej Załuski (Kraków), Tomasz Zieliński

Redakcja graficzno-techniczna:
Stefan Szczypka (kier.)
Małgorzata Luzińska
Beata Maruszewska

korekta: Maria Omiecińska, Romualda Miarecka

Wydawca: Krajowe Wydawnictwo Czasopism RSW "Prasa-Książka-Ruch", ul. Noakowskiego 14, 00-666 Warszawa, tel. centr. 25-72-91 do 93
Redakcja: ul. Mokotowska 48, 00-543 Warszawa, tel. 21-76-58 telex 815664 cestud pl (goście nas Warszawskie Centrum Studenckiego Ruchu Naukowego ZSP)
Skład i druk: Prasowe Zakłady Graficzne, Łódź, ul. Armii Czerwonej 28.
Cena: 100 zł Zam. 1390/86, P-59

Prenumerata: kwartalnie – 300 zł, półrocznie – 600 zł, rocznie – 1200 zł. Prenumeratę od instytucji przyjmują oddziały RSW, a od osób prywatnych poczta (na wsi także doręczyciele). Prenumeratę ze zleceniem wysyłki za granicę (droższą o 50% dla osób prywatnych i o 100% dla instytucji) przyjmuje Centrala Kolportażu RSW, ul. Towarowa 28, 00-958 Warszawa, NBP XV O/M W-wa 1153-201045-139-11.
Prenumerata przyjmowana jest na II, III i IV kwartał oraz na II półrocze z miesięcznym wyprzedzeniem, a na rok następny do 10 listopada.

Ogłoszenia przyjmuje Biuro Reklamy, ul. Mokotowska 5, tel. 25-35-36; adres dla korespondencji w sprawach ogłoszeń: Noakowskiego 14, 00-666 Warszawa. Zamawiając ogłoszenia listownie należy podać datę i miejsce wpłaty (konto KWCz: NBP III O/M W-Wa 1036-5294 z zaznaczeniem "ogłoszenie w KOMPUTERZE").

1 cm^2 ogłoszenia kosztuje 300 zł, najmniejsze ogłoszenie – 2100 zł, cała strona – 200 tys. zł; kolor dodatkowy – 30% drożej, pełna gama barw – 100% drożej. Za treść ogłoszeń redakcja nie odpowiada.

Nakład 100.000 egz.
Nr indeksu 36-345 ISSN 0860-2514

OCZEKIWANIA I DECYZJE

Przez długi, długi czas czekaliśmy. Najpierw na zainteresowanie informatyką, później, gdy mikrokomputery zdobyły serca (i uszczupliły kieszeń) dziesiątków, a obecnie już z pewnością setek tysięcy ich zwolenników, czekaliśmy na decyzje. Niektóre z nich już zapadły w pierwszych miesiącach br., pisaliśmy o tym w numerze 2.

Jedną z najnowszych decyzji jest porozumienie w sprawie dostawy sprzętu na rzeczywiście liczącą się skalę z firmą TIMEX.

Najważniejsze są jednak rozstrzygnięcia o znaczeniu strategicznym, wyznaczające kierunki rozwoju całych dziedzin gospodarki i polityki na najbliższe lata. Decyzje takie przyniosą uchwały X Zjazdu PZPR, a jesteśmy przekonani, że dzięki nim masowe zainteresowanie informatyką uzyska podstawy do okrzepnięcia i zamienienia się w konkretne efekty.

A zainteresowanie to jest olbrzymie. Niemały przecież nakład "Komputera" rozszedł się w ciągu kilku godzin, zaś redakcja została zmuszona do przyjęcia niewdzięcznej roli informatora, że "niestety, już nie ma". Podobne sukcesy są udziałem dodatków informatycznych do gazet codziennych: "Bajtka" i "IKS". Przez graniczne punkty kontrolne przewożone są codziennie dziesiątki różnorodnych komputerów. (Na szczęście bez cła! Wielka w tej mierze chwała prof. Iwo Białynickiemu – Biruli i Mikroklanowi w "Informatyce"!)

Skłania to do postawienia pytania: czy aby nie jest to tylko przemijająca moda? Niektórzy żurnaliści, widzący w mikrokomputerach jedynie pudełka do gier telewizyjnych i nic więcej, straszą nas od czasu do czasu hiobowymi wieściami w rodzaju: "krach na rynku mikrokomputerów w Wielkiej Brytanii!". Po czym okazuje się, że rynek ma się całkiem dobrze, do sprzedaży oferowane są coraz nowocześniejsze i tańsze modele, popyt utrzymuje się na wysokim poziomie, a cała wiadomość wynikła z chwilowego zastoju, kiedy to klienci nie byli jeszcze zdecydowani, co właściwie chcą kupić. Starsze modele sprzętu też zresztą znajdują nabywców, a należy przypuszczać, że znaczną ich część stanowią turyści ze Wschodu. Najśmieszniej jest jednak, gdy handlowcom z tamtej strony uda się sprzedać hurtowo większą ilość niezbyt udanego i nie produkowanego już modelu komputera.

Te handlowe rozważania prowadzą nas do wniosku, że "moda" na mikrokomputery wymusza coraz to nowe ich zastosowania, wyższą jakość sprzętu, a to z kolei powoduje dalsze zwiększenie ilości zastosowań mikrokomputerów.

Jesteśmy na samym początku tego cyklu, w fazie nasycania rynku sprzętem. Chociaż faza ta potrwa prawdopodobnie jeszcze

MENU:

dość długo, już teraz musimy przygotować się do fazy drugiej – szerokiego, konkretnego zastosowania mikrokomputerów nie tylko do zabawy, ale i do nauki i pracy. Takie zadanie stawia przed sobą i nasza redakcja.

Przed nami zaś faza trzecia – oparcie się na sprzęcie wyższej jakości, o różnorodnych możliwościach, ale na początek o przynajmniej częściowo ujednoliconych standardach. I tu trzeba koniecznie oprzeć się na krajowych wytwórcach (wyraźnie w liczbie mnogiej!), naszych, polskich programach, systemach i sieciach. Oczywiście nie wyłącznie, ale nikt pewnych rzeczy za nas nie zrobi, a nie w każdym przypadku można się uciec do starej, dobrej metody, nazwijmy to eufemicznie, kompilacji.

Do problemów i możliwości, które przyniesie omawiana druga i trzecia faza, trzeba przygotować i inżynierów, i humanistów. Czas, który do tej pory przeznaczaliśmy na czynności wykonywane znacznie szybciej i lepiej przez komputer, ten czas możemy obecnie przeznaczyć na myślenie i pracę twórczą. Stawia to całkiem nowe wymagania i zadania przed naukowcami, nauczycielami, urzędnikami, handlowcami, ale także przed rolnikami, prawnikami, lekarzami i artystami.

Uciec przed rewolucją informatyczno-komputerową się nie da, trzeba więc ją polubić i wykorzystać. Będzie ona oddziaływała i w sferze technologii i socjologii, wpłynie na życie każdego z nas. Żeby tę rzeczywistą rewolucję właściwie wykorzystać, potrzebna jest wiedza o teorii i zastosowaniach informatyki. Nasza redakcja liczy w tym względzie na Waszą pomoc i informacje.

Zakładamy redakcyjny bank danych o praktycznych zastosowaniach komputerów. Oczekujemy i przyjmujemy wszystkie zgłoszenia użytkowników prywatnych i służbowych, indywidualnych i zbiorowych. Prosimy o krótkie informacje o sprzęcie, pracy, którą wykonuje, zaletach i wadach przyjętego systemu. Sądzimy, że uda się w ten sposób stworzyć pierwszą polską kartotekę o informatyce w działaniu. W naszych zamierzeniach będzie ona pomocna i dla dzisiejszych użytkowników mikro- i minikomputerów (wymiana doświadczeń), i dla tych, którzy zechcą w przyszłości do tego grona dołączyć (porady).

Interesują nas wszystkie Wasze listy, chociaż nie możemy obiecać, że każdy otrzyma indywidualną odpowiedź. Cieszą nas szczególnie listy analizujące treść "Komputera" i zawierające propozycje nowych tematów. Taka korespondencja daje nam możliwość lepszego uwzględniania Waszych potrzeb, choć nie ze wszystkimi propozycjami możemy się zgodzić.

Wszystkim proponujemy współpracę autorską z redakcją, najlepsze materiały będą opublikowane. Artykuły przyjmujemy także w formie nagrań na kasetach magnetofonowych i dyskietkach, prosimy jednak o informacje dotyczące sposobu nagrania.

Dla tych, którym nie udało się kupić pierwszego numeru miesięcznika, a także dla wszystkich, którzy darzą nas sympatią lub niechęcią, powtarzamy wspólne hasło:

"MYŚL SAM, KOMPUTER CI POMOŻE!"

MAREK MŁYNARSKI

Wokół komputera

Choć staramy się – nie potrafimy i nie zdołamy zaspokoić wszystkich wobec nas oczekiwań, wyrażonych m.in. w listach, których drobną część publikujemy na str. 46 (w przyszłości rezerwujemy na ten cel m.in. i to miejsce). Głód informacji jest tak wielki, że wciąż trudno o podział zadań między nami i ukazującymi się wkładkami mikrokomputerowymi, od każdego oczekuje się informacji o wszystkim.

Pewnych rad i sugestii przyjąć nie tylko nie możemy, ale i nie chcemy. W telewizyjnym magazynie Spectrum, niegdyś znakomicie redagowanym przez Bogdana Borusławskiego, anonimowy recenzent po pobieżnym przekartkowaniu naszego pisma i skwitowaniu jego istotnych treści nic nie znaczącą uwagą "artykuł problemowy" (co może, jak widać, oznaczać i wywiad, i esej), namawiał nas na szersze zajęcie się komputerami innymi niż ZX Spectrum, np. Atari.

Nastąpiło tu pewne nieporozumienie: nie zajmujemy się żadnym konkretnym typem mikrokomputera, lecz problemami i metodami wspólnymi dla całej informatyki, ilustrowanymi jedynie przykładowymi programami. Ilustracja ta najczęściej przeznaczona jest dla właścicieli ZX Spectrum – bo po prostu w tej chwili stanowią oni ponad 70% użytkowników mikrokomputerów w Polsce (dane są wynikiem analizy ankiet zebranych wśród uczniów szkół warszawskich). Chcemy jednak, by nasze treści pozostały aktualne nawet wtedy, gdy pojawią się urządzenia przewyższające wszystkie znane dzisiaj.

Naszym głównym bohaterem, przedmiotem zainteresowania, nie jest komputer jako przedmiot i ładna zabawka do chwalenia się przed znajomymi, lecz jego zastosowania, niezależnie od typu użytego sprzętu. Pisać będziemy o światowych i polskich standardach technicznych. Dziś oznacza to Spectrum, Amstrada, Commodore 64 i IBM PC. W przyszłości zapewne także Atari 520 ST.

Niezależnie od tego mamy nadzieję, że użytkownicy różnego rodzaju mikrokomputerów, także Atari, założą wkrótce własne pisma klubowe, służące wymianie związanych z danym urządzeniem szczegółowych porad i informacji.

* * *

Redagując pierwsze numery "Komputera" musieliśmy podjąć kilka trudnych decyzji językowych. Bylibyśmy Wam wdzięczni za uwagi, czy zdecydowaliśmy słusznie. Postanowiliśmy pisać o manetce (w nawiasie: joystick), używać skrótu KB (przez duże K, bo chodzi tu o 1024 bajty) oraz używać rażącego ucho niefachowca, ale trudnego do zastąpienia sformułowania, że jakiś program pracuje POD systemem np. CP/M (a nie W systemie CP/M). Czy słusznie? W naszym przekonaniu tak jest najlepiej, ale być może mylimy się.

WŁADYSŁAW MAJEWSKI

Co, gdzie, kiedy

W marcu br. w Związku Radzieckim utworzono nowy Państwowy Komitet ds. techniki komputerowej i systemów informatycznych. Stojący na ponadministerialnym szczeblu organ koordynować ma proces komputeryzacji gospodarki narodowej i jest trzecią tego rodzaju instytucją powołaną od czasu objęcia przez Michaiła Gorbaczowa funkcji sekretarza generalnego KC KPZR. W ubiegłym roku powstały dwa takie państwowe komitety – rolno-przemysłowy, który przejął kompetencje kilku rozwiązanych ministerstw zajmujących się szeroko rozumianymi sprawami rolnictwa, oraz Państwowy Komitet ds. Budowy Maszyn.

Utworzenie specjalnego organu nadzorującego rozwój techniki komputerowej stanowiło ukoronowanie procesu przemian, którego pierwszym etapem było stworzenie w Nowosybirsku Centrum Obliczeniowego AN ZSRR oraz powierzenie tamtejszemu Syberyjskiemu Oddziałowi Akademii Nauk zadania opracowania i wdrożenia programu powszechnej edukacji informatycznej. Kolejne posunięcia, jakie poczyniono w ciągu ostatnich kilku lat – to utworzenie w Akademii Nauk ZSRR specjalnego oddziału informatyki, techniki obliczeniowej i automatyzacji, powołanie do życia Instytutu Problemów Cybernetyki oraz Instytutu Informatyki, a także Instytutu Problemów Technologii Mikroelektroniki i Materiałów Szczególnej Czystości. W skład tego ostatniego wchodzi również specjalne biuro konstrukcyjno-technologiczne oraz wydział produkcji doświadczalnej. Własne biuro konstrukcyjne posiada również utworzony ostatnio w Jarosławiu Instytut Mikroelektroniki.

Czego przemysł i szerzej – gospodarka radziecka oczekują od nowych placówek naukowych? Mówił o tym w wywiadzie udzielonym miesięcznikowi „Technika Mołodioży" wiceprezes Akademii Nauk ZSRR, akademik JEWGENIJ WIELICHOW: „Jednym z najważniejszych zadań tych instytutów jest stworzenie nowej techniki obliczeniowej. Myślę przede wszystkim o superkomputerach. Opracowanie projektu takich maszyn, wykonujących minimum 300 milionów operacji na sekundę, jest poważnym problemem. Chodzi wszak nie tylko o stworzenie nowej bazy układów scalonych o bardzo wielkiej skali integracji (VLSI) i rozpracowanie problemu odprowadzania od tych elementów ogromnych ilości ciepła, które powstaje przy tak dużej szybkości pracy, ale również o przygotowanie niezbędnego oprogramowania. Bez superkomputerów nie można już rozwiązywać poważnych zadań naukowych, wdrażać na masową skalę systemów automatycznego projektowania (CAD), dokonać postępu w dziedzinie projektowania i wytwarzania układów scalonych o bardzo wielkiej skali integracji. Odpowiednie ośrodki akademickie, a także Instytut Cybernetyki Akademii Nauk Ukraińskiej SRR we współpracy z ministerstwami przemysłowymi wykonały już niezbędną pracę, której celem było przygotowanie kilku perspektywicznych konstrukcji superkomputerów.

Instytut Problemów Informatyki pracuje nad innym typem komputerów – maszynami przygotowywanymi z myślą o masowym zastosowaniu. Przede wszystkim – nad małymi komputerami o dużej wydajności i mocy obliczeniowej. Wykorzystywane są one dziś w badaniach naukowych, w różnorodnych zautomatyzowanych systemach i zautomatyzowanych elastycznych ciągach produkcyjnych. Ten sam Instytut ma za zadanie udoskonalić radzieckie mikrokomputery i systemy małych elektronicznych maszyn obliczeniowych.

Stopniowo wypełniana jest istniejąca niegdyś luka w radzieckiej technice obliczeniowej. Tworzone są doskonałe komputery osobiste – uniwersalne i masowe, niezbędne do nauki studentom i uczniom, potrzebne inżynierom, a także w pracy biurowej. W tym wypadku ważne jest nie tylko jak najstaranniejsze przygotowanie samej konstrukcji maszyny, lecz także dostosowanie do niej odpowiedniego oprogramowania: będą wszak na niej pracować ludzie nie znający zasad tradycyjnego, matematycznego programowania.

Kilka innych ośrodków akademickich zajętych jest nadzwyczaj perspektywiczną pracą nad stworzeniem następnego pokolenia urządzeń mikroelektronicznych, bazujących na nowych efektach fizycznych i procesach technologicznych.

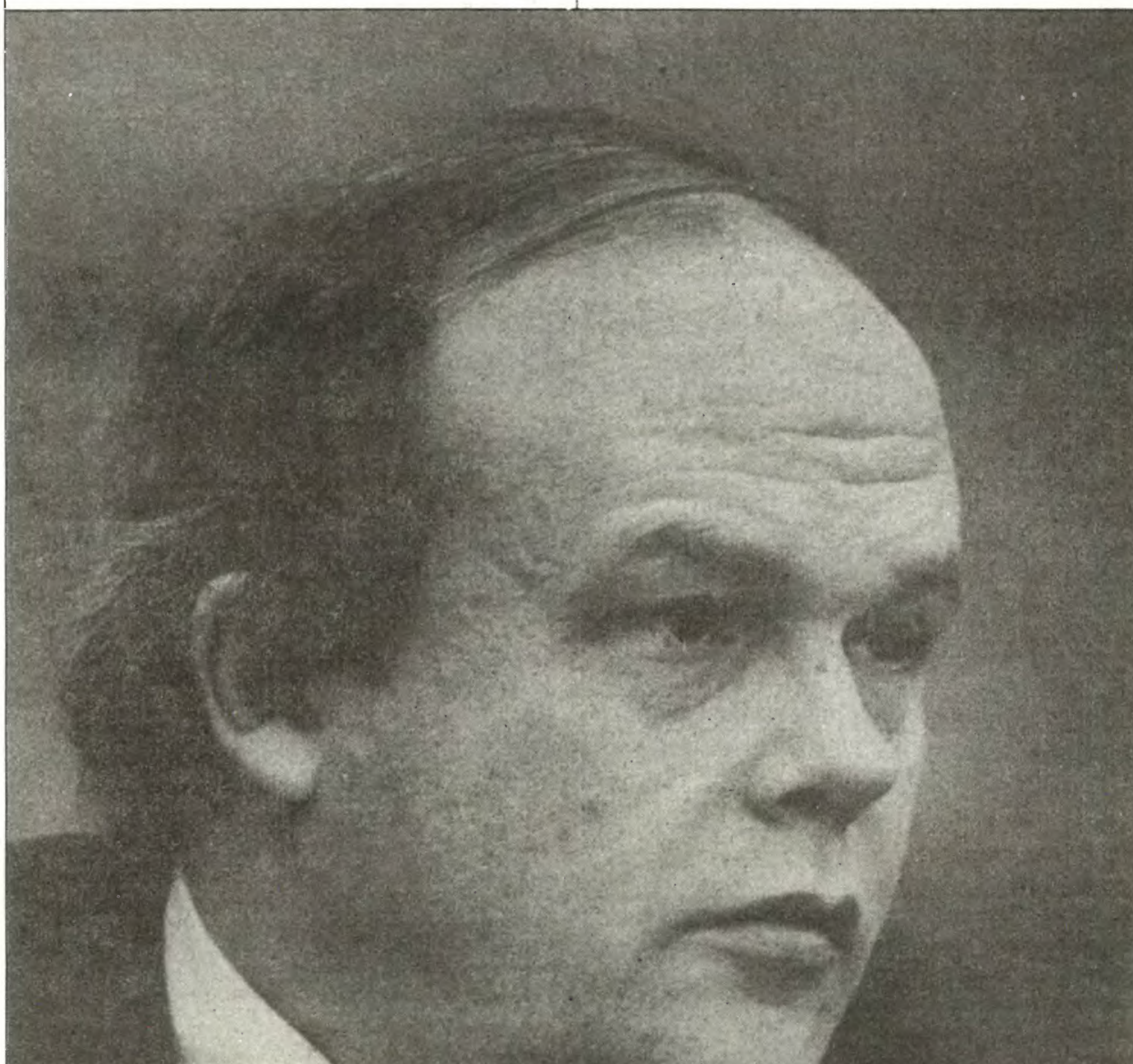
W obecnej, dwunastej pięciolatce w Związku Radzieckim rozpoczęta zostanie na masową skalę produkcja komputerów osobistych, z których korzystać będą nie tylko matematycy, lecz również konstruktorzy, technolodzy, lekarze, ekonomiści, architekci – a więc przedstawiciele różnorodnych zawodów, w tym również humaniści.

Równocześnie tworzony będzie system edukacji i reedukacji dziesiątków tysięcy ludzi, którzy korzystać będą z oprogramowania. Realizacja tego zadania wymagać będzie dużych nakładów i wielu wysiłków. Chodzi przecież nie tylko o wdrożenie nowych urządzeń i technologii, lecz również z gruntu odmiennych metod i nawyków pracy praktycznie we wszystkich gałęziach gospodarki narodowej.

Ustalono już kolejność realizacji tych ogromnych zadań. W pierwszym etapie, w ciągu 4-5 lat w wiodących gałęziach przemysłu, w Akademii Nauk, w wyższych uczelniach, będą tworzone ośrodki naukowo-produkcyjne ds. programowania. Każdy z nich zatrudni ok. 100-150 wysoko wykwalifikowanych specjalistów, władających współczesną "technologią" programowania. Przygotowaniem kadry dla tych ośrodków zajęły

---

Akademik J. Wielichow



się wydziały matematyki i matematyki stosowanej wiodących radzieckich uczelni i uniwersytetów. Myślę, że 25-30 takich ośrodków programowania rozwiąże zadania, jakie stoją przed nami w chwili obecnej.

Równocześnie tworzona jest ogólnokrajowa sieć ośrodków edukacji i reedukacji programistów, dla których programowanie będzie pracą nie naukową, a stosowaną, czysto produkcyjną.

W drugim etapie realizacji radzieckiego programu rozwoju i wykorzystania techniki obliczeniowej, w każdej gałęzi gospodarki narodowej, powstanie własna sieć takich ośrodków naukowo-produkcyjnych. Według wspólnego, gałęziowego programu realizować one będą zarówno problemy naukowe, jak i produkcyjne. Oczywiście jeden z takich ośrodków będzie wiodący. Określi on cały program prac w swojej gałęzi i zajmie się ich koordynacją. Koordynacja międzygałęziowa spocznie na barkach AN ZSRR.

Na bazie każdego wiodącego ośrodka gałęziowego stworzy się specjalne pododdziały. Ich zadaniem będzie edukacja i reedukacja specjalistów, korzystających z oprogramowania. Ludzie, którzy ukończą taką szkołę, zajmą się właśnie komputeryzacją produkcji, systemem zarządzania oraz nauki i oświaty wyższego i średniego stopnia.

Jest rzeczą oczywistą, iż takie pododdziały muszą jak najściślej współpracować z wyższymi uczelniami zajmującymi się kształceniem specjalistów dla danej gałęzi przemysłu. Pomogą im w procesie wdrażania komputerów do praktyki oświatowej. Efektem tej pracy będzie stworzenie infrastruktury nowej gałęzi gospodarki narodowej – przemysłu zajmującego się produkcją oprogramowania.

Dla skomputeryzowania każdego ogniwa łańcucha – "technika-produkcja", utworzono trzy ośrodki opierające się na istniejącym w Nowosybirsku Ośrodku Obliczeniowym AN ZSRR.

Pierwszy, wspierany przez Instytut Fizyki Teoretycznej im. Ł. Landau'a, umożliwia fizykom-teoretykom pracę na najnowocześniejszym sprzęcie komputerowym. Drugi ośrodek jest ściśle związany z ministerstwami przemysłu elektrotechnicznego i przemysłu środków łączności. Jego głównym zadaniem jest stworzenie wespół z wieloma instytutami badawczymi automatycznego systemu projektowania układów scalonych VLSI. System ten musi mieć potężne oprogramowanie, umożliwiać modelowanie, testowanie i dialogową obróbkę informacji. Wreszcie trzeci ośrodek, utworzony przez Uniwersytet Moskiewski i Moskiewską Fabrykę Samochodów im. Lichaczowa, zajmuje się problemami automatyzacji projektowania w dziedzinie budowy maszyn.

Chciałbym podkreślić, iż tworzenie takich ośrodków jest jednym z elementów naszej polityki zmierzającej do umocnienia więzi nauki z przemysłem. Stosunkowo niedawno przyjęta została uchwała KC KPZR i Rady Ministrów ZSRR o tworzeniu międzygałęziowych kompleksów naukowo-technicznych. W ich skład wchodzić będą ośrodki naukowe, organizacje techniczne, biura konstrukcyjne i przedsiębiorstwa z różnych gałęzi przemysłu. Taki właśnie jeden z pierwszych międzybranżowy kompleks naukowo-badawczy, powołany w ramach AN ZSRR, zajmuje się problematyką laserów technologicznych. Na podkreślenie zasługuje fakt, iż wszystkie te kompleksy pracować będą według wspólnego planu.

Nie ulega wątpliwości, iż przedstawiony porządek prac nad głównymi kierunkami postępu naukowo-technicznego umożliwi przełamanie podziałów branżowych, przyczyni się do dalszego umocnienia więzi nauki z przemysłem".

Opr. HALINA MADEJCZYK

CP/M 86

W pierwszych numerach "Komputera" omówiliśmy systemy UNIX i CP/M. Dziś przedstawiamy wersję CP/M dla mikrokomputerów 16-bitowych z mikroprocesorem rodziny INTEL 8086, skoncentrujemy się przy tym na różnicach między CP/M 86 i CP/M. W następnych numerach zajmiemy się systemem PC DOS i jego pochodnymi.

BUDOWA I ORGANIZACJA SYSTEMU

CP/M 86 jest w pełni kompatybilny ze swoim 8-bitowym poprzednikiem CP/M 2.2. Umożliwia to przenoszenie programów w sposób automatyczny, przy minimalnym wysiłku programisty, a w konsekwencji udostępnia w CP/M 86 bogatą bibliotekę programów napisanych dla CP/M.

Jest to możliwe dzięki bardzo podobnej organizacji obu systemów. Różnice między nimi wynikają głównie z faktu rozszerzenia przestrzeni adresowej procesora 16-bitowego do 1 MB (64 KB w procesorze 8-bitowym). Schemat mapy pamięci ukazany jest na rys. 1.

CP/M 80 nie umożliwia relokacji programów. Są one ładowane pod stały adres (100_H). W związku ze zmienną wielkością modułu BIOS oraz możliwością wykorzystania obszaru zajętego przez moduł CCP, CP/M 80 ładowany jest w górny obszar pamięci.

Natomiast CP/M 86 jest z założenia relokowalny. Zmiana zawartości rejestrów segmentowych procesorów 8086 lub 8088 automatycznie zmienia wszystkie adresy. Dzięki temu zarówno system operacyjny, jak i programy mogą być umieszczane w dowolnym miejscu pamięci za wyjątkiem obszaru zajętego przez wektor przerwań (adresy $00000_H - 00400_H$). Zazwyczaj CP/M 86 jest ładowany bezpośrednio ponad tym obszarem.

W związku z relokowalnością systemu CP/M 86 nie może używać adresów absolutnych dla wejść systemowych lub parametrów domyślnych. Tak więc wejście do modułu BDOS odbywa się poprzez zarezerwowane na to przerwanie programowe nr 224 (w CP/M 80 poprzez wykonanie instrukcji CALL 0005_H), zaś do modułu BIOS poprzez dodatkową funkcję BDOS nr 50 (w CP/M 80 przez bezpośrednie odwołanie się do tabeli skoków adresowanej w sposób absolutny).

MODUŁ CCP (CONSOLE COMMAND PROCESSOR)

Jest to część systemu operacyjnego zajmująca się przetwarzaniem poleceń operatora. Umożliwia wykonanie poleceń rezydentnych (stale obecnych w pamięci) lub ładowanie i wykonanie poleceń nierezydentnych (przechowywanych na dysku) oraz programów użytkowych.

W CP/M 86 obszar zajęty przez moduł CCP w pamięci operacyjnej nie może być wykorzystany jako obszar danych programu użytkowego. Po inicjacji systemu wszystkie moduły są na stałe obecne w pamięci.

MODUŁ BDOS (BASIC DISK OPERATING SYSTEM)

Jest to serce systemu CP/M. Moduł ten realizuje wszystkie żądania systemu operacyjnego zamieniając je w proste żądania na poziomie sprzętowym i przekazując je do modułu BIOS.

Z punktu widzenia użytkownika moduł ten to zbiór funkcji realizujących określone zadania. W celu przekazania parametrów zarówno wejściowych, jak i wyjściowych w CP/M 86 używane są rejestry procesora zgodnie z następującą konwencją:

- rejestry wejściowe:
  - -CL : jednobajtowy numer funkcji BDOS,
  - -DL : jednobajtowy parametr,
  - -DX : parametr o długości 1 słowa lub 2 bajtów,
  - -DS : adres bazowy segmentu danych,
- rejestry wyjściowe:
  - -AL : jednobajtowe wartości (najczęściej kod błędu),
  - -AX i BX : wartość o długości 1 słowa lub 2 bajtów,
  - -ES i BX : wartość o długości 2 słów, gdzie: BX zawiera adres względny w segmencie, ES zawiera adres bazowy segmentu.

Tabela 1 zawiera dokładny wykaz funkcji BDOS. Gwiazdką oznaczone są funkcje nowe, nie istniejące w CP/M 80 lub których działanie zostało zmienione w stosunku do wersji 2.2 tego systemu.

Wszystkie funkcje modułu BDOS można podzielić na następujące grupy:

– (0-12) : funkcje proste – organizują one prostą komunikację z logicznymi urządzeniami zewnętrznymi, takimi jak konsola, drukarka, czytnik i perforator taśmy papierowej. Działanie jedynie 2 funkcji z tej grupy zostało nieco rozszerzone w stosunku do CP/M 2.2.

Pragnę zwrócić uwagę na funkcję nr 0 – zerowanie systemu. W CP/M 86 możliwe są 2 rodzaje zerowania zależnie od zawartości rejestru DL:

DL=00_H – działanie programu jest przerywane, jego obraz nie pozostaje dłużej w pamięci.

DL=01_H – działanie programu jest również przerywane, ale zawartość pamięci operacyjnej pozostaje bez zmian. Program można powtórnie uruchomić od punktu przerwania.

▶ 6

5 ◀

W CP/M 80 wykonanie funkcji 0 jest zawsze równoważne ponownemu załadowaniu systemu do pamięci operacyjnej.

– (13 – 40, 51 – 52) : funkcje dyskowe – są to najbardziej skomplikowane funkcje tego modułu. Ich zadaniem jest organizacja współpracy systemu ze stacjami dyskowymi. Począwszy od wersji CP/M 2.0 wszelkie operacje na dyskach zostały sparametryzowane przy użyciu tzw. tabel dyskowych. Tabele te umożliwiają współpracę z maksymalnie 16 stacjami dyskowymi o różnych parametrach. W celu łatwej implementacji nowych urządzeń dyskowych tabele te zostały umieszczone w module BIOS i są dostępne programistom.

Zmiany istniejących i dodanie nowych funkcji tej grupy spowodowane były rozszerzeniem przestrzeni adresowej oraz sposobem obliczania adresu efektywnego. Jedynie zmiana w działaniu funkcji nr 37 (inicjacja stacji dyskowej), polegająca na zerowaniu rejestru AL, spowodowana jest zapewnieniem kompatybilności systemów CP/M z wieloużytkownikowymi systemami MP/M;

– (53 – 59) : funkcje zarządzające pamięcią – są to funkcje nowe, nie występujące w CP/M 80. W systemie tym programy są ładowane w ciągłe obszary pamięci i nie ma żadnej możliwości kontroli nad sposobem ich rozmieszczenia.

CP/M 86 dzięki innej organizacji pamięci i istnieniu odpowiednich funkcji BDOS jest pod tym względem dużo bardziej elastyczny.

Niektóre systemy komputerowe mogą posiadać kilka nieciągłych lub niezależnych regionów pamięci. Musi więc istnieć mechanizm programowy informujący system operacyjny (w tym wypadku BDOS) o ich liczbie, rozmieszczeniu i rozmiarze. W CP/M 86 dane te są zakodowane w tabeli regionów pamięci MRT (Memory Region Table) umieszczonej w module BIOS.

Gdy system operacyjny zna fizyczną organizację pamięci, BDOS może rozmieścić w niej programy lub segmenty podczas ładowania. Zapewnione jest to dzięki dodaniu specjalnej funkcji BDOS (nr 59).

– (50) : bezpośrednie wywołanie modułu BIOS. Funkcja ta została dołączona w CP/M 86 w celu zapewnienia relokowalności systemu operacyjnego.

MODUŁ BIOS (BASIC INPUT OUTPUT SYSTEM)

Ta część systemu operacyjnego jest odpowiedzialna za przeprowadzenie wszystkich operacji na fizycznych urządzeniach zewnętrznych. Na niej skupia się cała zależność systemu od konkretnej konfiguracji sprzętowej. Moduł ten jest wymienny (inny dla różnych implementacji). Ścisłe zdefiniowanie wszystkich funkcji BIOS ułatwia uruchomienie tego modułu na nowych mikrokomputerach i umożliwia łatwą przenoszalność systemu CP/M.

Wejście do modułu BIOS odbywa się za pośrednictwem specjalnej funkcji BDOS. Ta przekazuje sterowanie do odpowiedniego miejsca w wektorze skoków BIOS umieszczonego począwszy od adresu 2500_H względem bazy systemu operacyjnego. Wektor ten jest sekwencją 3-bajtowych instrukcji skoków, które z kolei przenoszą sterowanie do poszczególnych punktów wejścia tego modułu. Procedury tam umieszczone realizują funkcje BIOS.

POLECENIA

Istnieją następujące typy poleceń systemu CP/M:

– rezydentne, stanowiące część systemu operacyjnego i razem z nim ładowane do pamięci;

– nierezydentne, ładowane do pamięci przy każdorazowym odwołaniu do nich. Są one umieszczone na dysku systemowym jako plik wynikowy typu . CMD.

Organizacja poleceń w CP/M 86 jest taka sama, jak w CP/M 80. Jedynie nieznacznie zwiększono ich ilość.

Badano następujące polecenia:

– DIRS – umożliwia wyprowadzenie opisów plików systemowych;

– COPYDISK – umożliwia zrobienie dokładnej kopii (sektor po sektorze) wskazanej dyskietki;

– HELP – powoduje wyprowadzenie pełnego zestawu poleceń (zarówno wbudowanych jak i nie);

– TOD (Time of Day) – umożliwia przypisanie plikom daty i dokładnego czasu. Może to być np. termin powstania pliku.

OCENA SYSTEMU

CP/M 86, jak do tej pory, nie odegrał poważniejszej roli na rynku 16-bitowych systemów operacyjnych. Myślę, że złożyły się na to następujące przyczyny:

– w momencie wypuszczenia CP/M 86 przez firmę DRI był już obecny na rynku system MS-DOS firmy MICROSOFT;

– firma IBM mając gotowy swój pierwszy komputer osobisty IBM PC, który ustanowił później standard na mikrokomputery 16-bitowe, podpisała porozumienie z MICROSOFT na dostarczenie systemu operacyjnego podobnego do MS-DOS o nazwie PC-DOS;

– okazało się, że przeniesienie schematów rozwiązań z systemów 8-bitowych przy minimalnych i jedynie koniecznych zmianach związanych z inną architekturą procesora to trochę za mało jak na zupełnie inną klasę mikrokomputera.

Jednak system CP/M 86 jest ciągle obecny na rynku systemów operacyjnych. Zawdzięcza to swoim następującym zaletom:

– pełnej kompatybilności z systemem CP/M 2.2, a dzięki temu możliwości korzystania z bogatej biblioteki tego systemu;

– łatwości instalacji systemu na nowych mikrokomputerach;

– możliwości takiej instalacji już przy 64 KB pamięci operacyjnej (dla porównania – minimalna konfiguracja pamięci dla systemu PC-DOS to 128 KB).

System ten ma jednak dość istotne wady. Oto niektóre z nich:

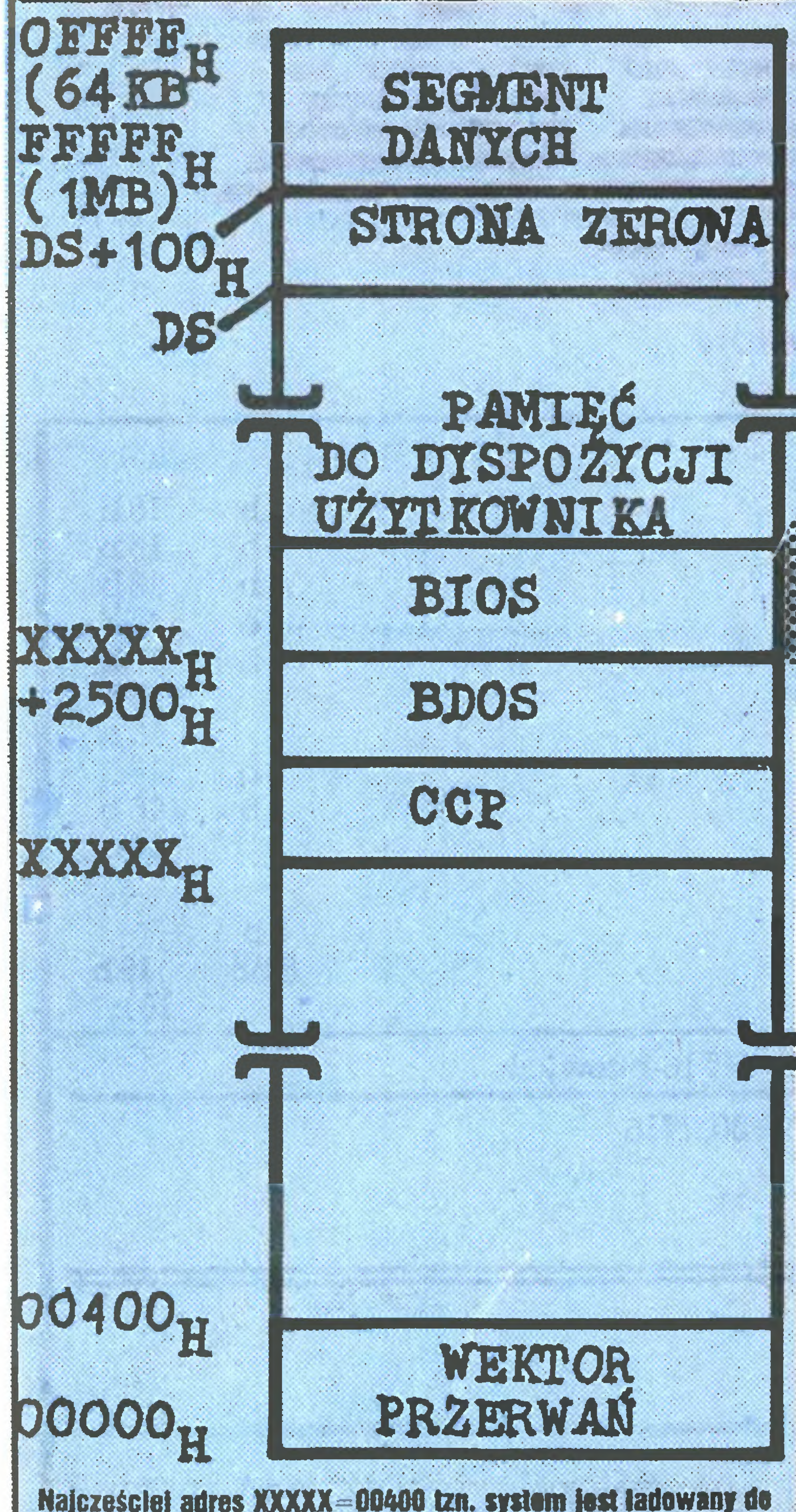
– brak możliwości ochrony plików;

– brak podprogramów sortowania zbiorów danych;

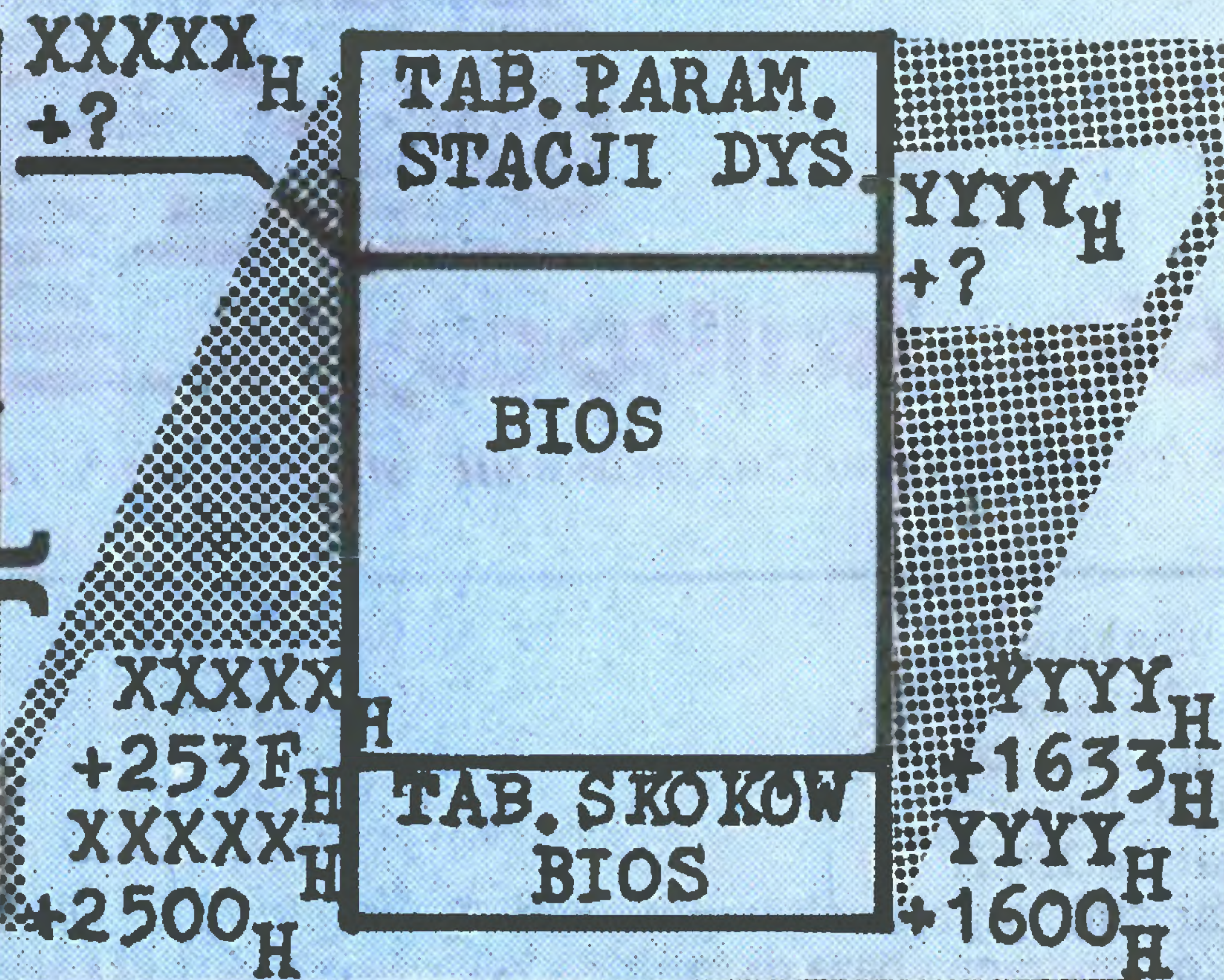
– brak indeksowo-sekwencyjnej organizacji katalogu;

FUNKCJE MODUŁU BDOS

Nr funkcji	Funkcja
1	2
0*	zeruj system
1	wprowadź znak z konsoli
2	wyprowadź znak z konsoli
3	wyprowadź znak z czytnika
4	wyprowadź znak na perforator
5	wyprowadź znak na drukarkę
6*	bezpośrednia komunikacja z konsolą
7	pobierz IOBYTE
8	ustaw IOBYTE
9	wydrukuj łańcuch znaków
10	wczytaj bufor konsoli
11	sprawdź stan konsoli
12	wyprowadź numer wersji systemu
13	wyzeruj system dysków
14	wybierz dysk
15	otwórz plik
16	zamknij plik
17	szukaj pierwszego pliku
18	szukaj następnego pliku
19	usuń plik
20	czytaj sekwencyjnie
21	pisz sekwencyjnie
22	utwórz plik
23	zmień nazwę pliku
24	pobierz wektor LOGIN
25	pobierz numer aktualnego dysku
26	ustaw adres DMA
27*	pobierz adres wektora alokacji
28	zapisz chroniony dysk
29	pobierz wektor R/O
30	ustaw atrybuty pliku
31*	pobierz adres DPB
32	ustaw/pobierz kod
33	czytaj rekord z pliku o dostępie bezpośrednim
34	pisz rekord do pliku o dostępie bezpośrednim
35	oblicz rozmiar pliku
36	ustaw bieżący numer rekordu
37*	inicjuj stacje dyskowe
40	pisz losowo z dopełnieniem zerami
50*	wywołaj moduł BIOS
51*	ustaw bazę DMA
52*	pobierz bazę DMA
53*	pobierz maksymalny dostępny blok pamięci
54*	pobierz maksymalny blok pamięci od adresu absolutnego
55*	przydziel region pamięci
56*	przydziel region pamięci od adresu absolutnego
57*	zwolnij region pamięci
58*	zwolnij wszystkie regiony
59*	załaduj program



Najczęściej adres XXXXX=00400 tzn. system jest ładowany do pamięci bezpośrednio ponad obszar zarezerwowany na wektor przerwań. Strona zerowa zajmuje pierwsze 128 bajtów segmentu danych.



– z góry ograniczona ilość pozycji w katalogu, a przez to utrudniona współpraca z dyskami stałymi;

– brak możliwości równoczesnego drukowania plików i wykonywania innych poleceń;

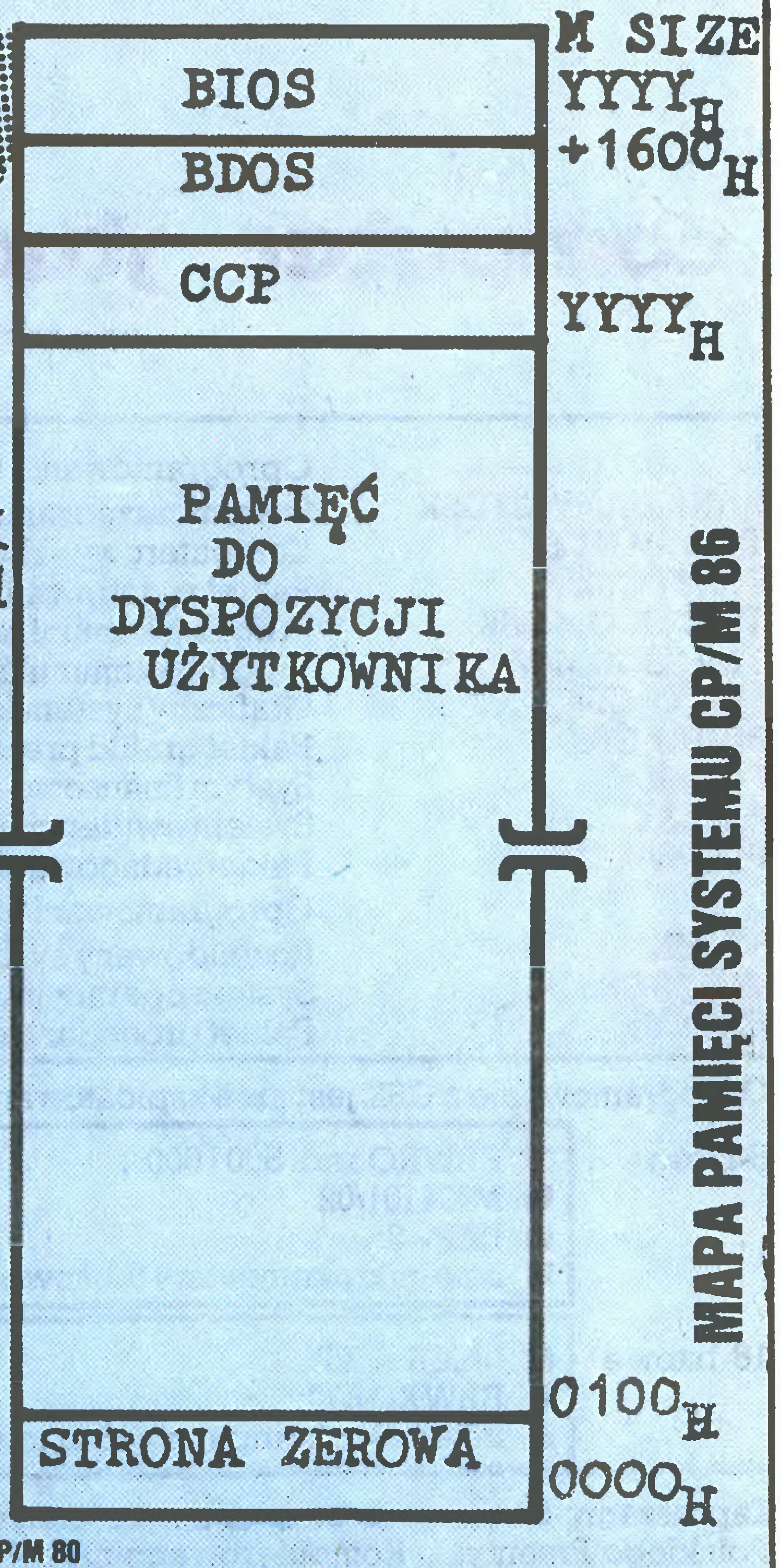
– brak możliwości weryfikacji informacji zapisanych na dysku;

– ograniczona ilość stacji dyskowych, które można podłączyć do systemu.

DANUTA MAGDZIK

Polecamy następujące książki:

1. M. Sobczyk: Dyskowy System Operacyjny CP/M, Informatyka, 2, 3, 4, 1983.
2. M. Dahmke: The Byte Guide to CP/M 86, USA 1984.



MAPA PAMIĘCI SYSTEMU CP/M 86

CP/M 80
Dla wersji 2.2 tego systemu
20 KB < MSIZE < 64 KB.
YYYY oznacza bliżej nieokreślony adres.

"...KOMPUTEREM DO KLIENTA!"

Przedsiębiorstwo Techniki Obliczeniowej i Organizacji (KSH) otworzyło w Budapeszcie nowe biuro. Mieści się ono w stylowym secesyjnym budynku i oferuje najnowsze osiągnięcia techniki komputerowej. Zostało ono powołane z myślą o wszystkich, począwszy od uczniów szkół podstawowych, a na profesjonalistach skończywszy.

Głównym celem biura jest propagowanie rozmaitych form zastosowania komputerowej techniki obliczeniowej i położenie fundamentów pod sieć powszechnych, komputerowych placówek handlowych.

Przeciętnym indywidualnym klientom biuro oferuje mikrokomputery, oprogramowanie i informacje o nich. Uruchomiono 6 stanowisk pracy wyposażonych w mikrokomputery osobiste i profesjonalne. Klienci zupełnie początkujący mogą skorzystać z porad, których udzielają im specjaliści zatrudnieni w biurze: informatycy, elektronicy, inżynierowie i ekonomiści. Ci, którzy zechcą zrobić coś samodzielnie, mogą skorzystać z samych urządzeń, oczywiście po uiszczeniu odpowiedniej opłaty: 170 forintów za godzinę pracy na sprzęcie profesjonalnym i 100 forintów za korzystanie z Commodore 64. Uczniom i studentom udzielane są zniżki.

Do tej nietypowej placówki usługowej można przyjść z własnymi programami, ale także można zakupić je na miejscu. Zestaw programów (10 sztuk) kosztuje ok. 1200 forintów. Dostępny jest również papier do drukarek – 1000 kartoników w cenie 470 forintów. Artykuł ten jednak sprzedawany jest w ograniczonych ilościach, tylko użytkownikom indywidualnym.

Na miejscu można również zakupić poradniki i książki. Oferta obejmuje 50 pozycji.

Do tego komputerowego salonu mogą zgłaszać się także przedsiębiorstwa. Udziela się im bezpłatnych konsultacji, natomiast już odpłatnie można zbadać możliwości zastosowania informatyki w rozwiązaniu konkretnych problemów danego przedsiębiorstwa.

W salonie można nabyć mikrokomputery M08X i TAP 34, wraz z oprogramowaniem. Do każdego zakupionego aparatu dołącza się 10 programów. W najbliższej przyszłości zamierza się połączyć sklep z centrum komputerowym Przedsiębiorstwa Techniki Obliczeniowej i Organizacji, gdzie napływające dane będą przetwarzane przez komputery R 35, IBM i Siemens.

Biuro pełni także rolę salonu wystawowego: każde przedsiębiorstwo może zwrócić się z prośbą o prezentację swego sprzętu.

Organizatorzy placówki w Budapeszcie planują otwarcie kolejnych biur – w Kocskemét i Szombathely. Z czasem tego rodzaju salony powstaną przy wszystkich terenowych delegaturach Przedsiębiorstwa Techniki Obliczeniowej i Organizacji.

Dla "Komputera" opracował:
Szczepan Woronowicz
wg Sára Zárda,
Computer-M,
Mikroszamitógép Magazin
Nr 3/1985



ENIAC – CZTERDZIESTOLATEK NA EMERYTURZE

W połowie lutego br. odbył się bankiet z okazji czterdziestolecia narodzin ENIACA, pierwszego na świecie elektronicznego komputera. Była to maszyna wprost rewelacyjna: liczyła 1000 razy szybciej niż najlepsze poprzednie maszynki elektromechaniczne! To, co zajęłoby 40 osobodni liczenia na wielkich, hałaśliwych kalkulatorach – ENIAC liczył w 20 sekund.

Nazwa "pradziadka" jest skrótem od Elektronic Numerical Integrator and Computer – Elektroniczny Cyfrowy Przyrząd Całkujący i Obliczeniowy. Składał się z 40 modułów, a każdy mieścił się w czarnej metalowej szafce wysokiej na ponad 2,70 m. Łącznie ważyły one 30 ton. Były usiane oscyloskopami, przełącznikami, światełkami kontrolnymi i pracowały na lampach elektronowych (teraz tylko starsi ludzie pamiętają te wielkie lampy radiowe...). W pomieszczeniu, w którym stał ENIAC – wielkości sporego garażu – musiały pracować wentylatory przemysłowe, bowiem lampy wydzielały tyle ciepła, że bez intensywnej wymiany powietrza wnętrze nagrzałoby się tak silnie, że przewody rozlutowałyby się albo wręcz stopiły przy dłuższych obliczeniach.

Programowanie – trwające czasami i dwa dni – polegało na ręcznym przestawianiu na dziesiątkach szaf setek, jeśli nie tysięcy przełączników.

Technicy dopomagali sobie mnogością kabli z wtyczkami, co w efekcie przypominało do złudzenia spotykane jeszcze w Polsce ręczne centralki telefoniczne.

Dane wprowadzano na kartach perforowanych. Przy pierwszym praktycznym zastosowaniu użyto ich milion! Chodziło o obliczenia związane z najściślej dotąd tajną bronią – nie testowanym wtedy jeszcze typem bomby wodorowej.

W funkcjonowaniu ENIACA były dwie zmory dla techników. Pierwsza – to przepalanie się przynajmniej dwa razy dziennie jednej z 18 000 lamp. Trzeba było mozolnie szukać tej, którą należało wymienić. Druga – to myszy; przynajmniej do czasu, gdy naukowcy nie złapali małego stadka, które po kilkudniowym przegłodzeniu poczęstowano kawałkami różnych izolacji. Te rodzaje, które myszom smakowały – starannie usunięto z komputera.

ENIACA stworzył 500-osobowy zespół techników pracujących pod kierunkiem J. Prespera Eckerta i nieżyjącego już Johna Mauchly. Ludzie pracowali zupełnie w stylu Krzemowej Doliny – z ogromnym zaangażowaniem, niemal w euforii tworzenia czegoś naprawdę rewolucyjnego. U Jobsa, w Apple Computer, dla zespołu konstruującego McIntosha przed dwoma laty dostarczano napoje chłodzące za darmo i masażystę, bo pracowano po 12-16 i niekiedy 24 godziny na dobę. Przy ENIACU zaś w pokoju obok stały dla najbardziej wyczerpanych dwie leżanki. Czasy się zmieniły i pod tym względem, o genialnych projektantów dba się teraz znacznie lepiej.

Eckert i Mauchly nie zrobili wielkiej kariery. W miesiąc po uruchomieniu komputera zrezygnowali z pracy w Moore School of Electrical Engineering, bowiem nie chcieli się dzielić z uczelnią prawami do patentu. Założyli firmę, która pierwsza na świecie w 1951 r. zaoferowała do zastosowań handlowych komputer UNIVAC 1. Przeciwności losu sprawiły, że bankrutującą firmę wynalazcy musieli sprzedać koncernowi Remington Rand. Ale prawdziwa klęska nadeszła w 1973 r., gdy po wieloletnim procesie o prawa patentowe firma Honeywell uzyskała wyrok stwierdzający, że Eckert i Mauchly spożytkowali jedynie pomysł Johna Atanaseffa. Przestały im przysługiwać płacone przez liczne firmy opłaty za korzystanie z wynalazku, odebrano im także moralne prawa do niego.

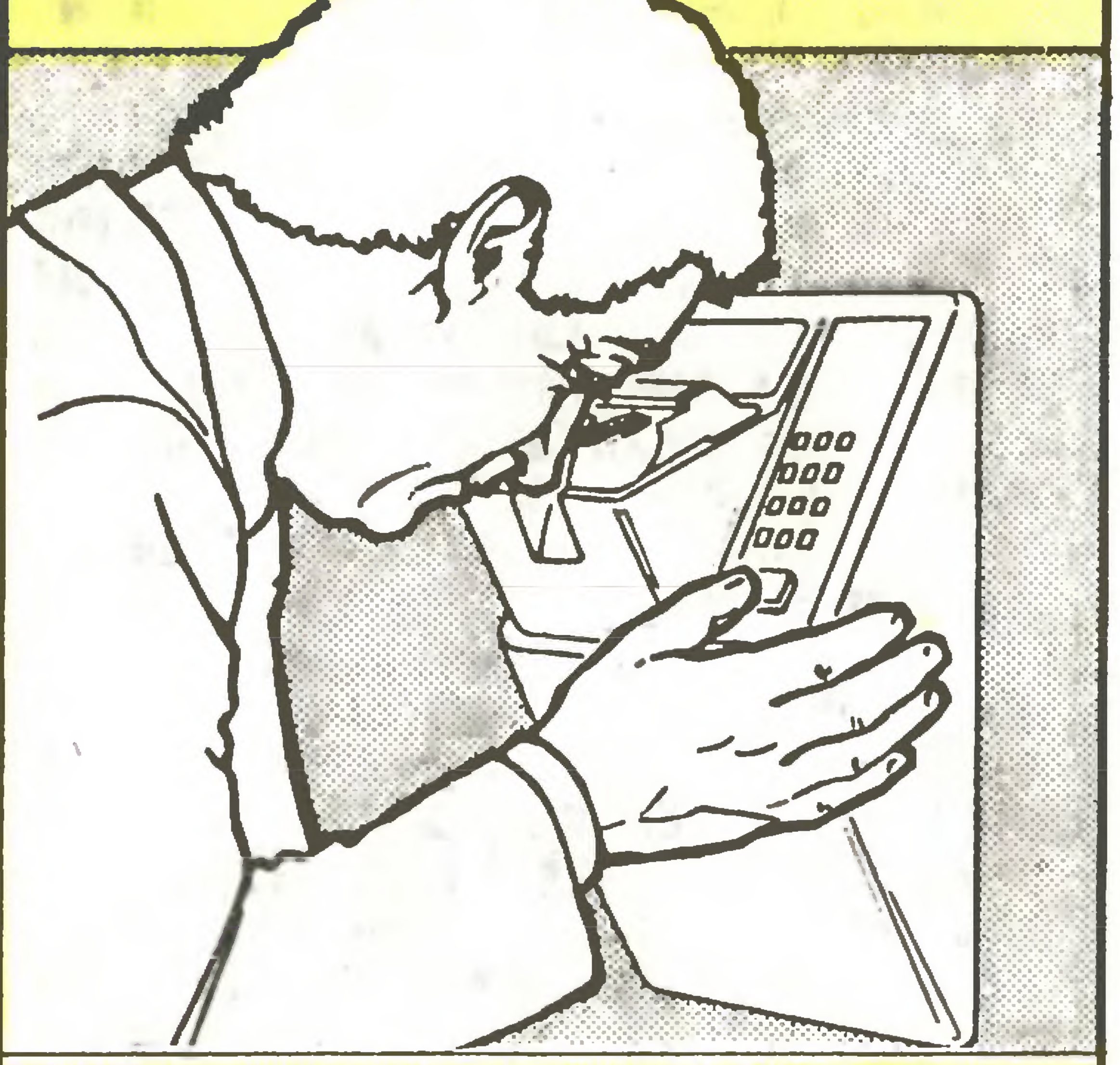
Tymczasem ENIAC funkcjonował przez prawie 10 lat, mozoląc się nad obliczeniami wojskowymi. Gdy zakończył służbę, do uczelni, gdzie powstał, powróciły cztery moduły, inne trafiły do różnych muzeów techniki. Tam najczęściej obok tych wielkich szaf umieszczona jest mała gablotka, w której przy pomocy szkła powiększającego można dokładnie obejrzeć nowoczesną kostkę krzemową, która ma moc obliczeniową porównywalną z ENIACEM, ale za to mieści się na paznokciu.

(JAL)

SPÓJRZ MI W OCZY TO CIĘ PUSZCZĘ

Gdy na początku zeszłego roku opracowano kampanię reklamową "Coca-Coli" o nowej recepturze, w agencji reklamowej wszelkie z tym sprawy wolno omawiać było tylko w sali konferencyjnej specjalnie zabezpieczonej przed wszystkimi znanymi metodami podsłuchu. Notatki pisano ręcznie, by obyć się bez maszynistek. Całego kramu pilnował uzbrojony strażnik, który wpuszczał znane sobie osoby z odpowiednimi kartami identyfikacyjnymi. To nie szpiegomania: gra toczyła się o zwiększenie udziału w sprzedaży napojów chłodzących na świecie, zatem o wiele miliardów dolarów.

Ale strażnika można przekupić, pobić, zabić lub odesłać z "ważnym poleceniem". Klucze można zabrać przemocą, ukraść czy dorobić.



Zamki cyfrowe, malutkie klawiaturki, na których wystukuje się szyfr, są już przestarzałe w porównaniu z tym, co usiłują zrobić dla różnych firm i instytucji chroniących swe tajemnice spece od elektroniki.

Zwrócono się w stronę biometrii, bowiem nawet najgłupszy strażnik rozpoznaje człowieka po tysiącu cech jednocześnie, ale komputery póki co muszą się ograniczyć co najwyżej do kilku.

Wymyślono już systemy, w których trzeba wypowiedzieć parę słów do mikrofonu, a cechy głosu porównywane są ze wzorcem ("Sezamie otwórz się"?). Inne wymagają złożenia podpisu na specjalnej płytce, jeszcze inne przyłożenia dłoni (brane są wtedy wszystkie wymiary). Listę rozwiązań uzupełnia daktyloskopia – komputerek porównuje odciski palców. Ale głos może być zniekształcony chrypką czy emocjami, linie papilarne uszkodzone ranką, blizną lub zamazane brudem. Poza tym wzorce opisanych tu cech zajmują po około 250 bajtów pamięci. Dlatego amerykańska firma EyeDentify uważa, że jej system jest najlepszy: gdy zajrzy się w specjalny wizjer (jak w dawnych fotoplastykonach – któż to jeszcze pamięta?!) – komputer porównuje ze wzorcem siateczkę naczyń krwionośnych na tęczówce oka, co daje się opisać tylko 40 bajtami. Jeśli porównanie zgadza się – sezam staje otworem. Eksperci twierdzą, że istnieje jedna szansa na milion wpuszczenia na salę tajnych obrad osoby, której wprawdzie dobrze, tj. według wzorca, z oczu patrzy, ale nie mającej prawa wstępu. Podobnie – prawdopodobieństwo, że nie wpuści się tego, kogo powinno – jest jak 1:1000.

Cena urządzenia – wraz z zamkiem – wynosi 11000 dolarów.

(JAL)

Popularność LOGO wciąż rośnie. Wszystkie liczące się na światowym rynku mikrokomputery są obecnie firmowo wyposażane w translator tego języka. Niestety w naszym kraju dostępne są jedynie książki o języku BASIC, języku instalowanym fabrycznie w najprostszych mikrokomputerach.

LOGO doskonale wyrabia intuicję informatyczną, a jako dialekt LISP'u używane jest do rozwiązywania skomplikowanych problemów, m.in. przetwarzania języka naturalnego i sztucznej inteligencji.

LOGO budzi także zainteresowanie nauczycieli, choć nadal wielu z nich trwa przy BASIC'u.

W "Przeglądzie Technicznym" i telewizyjnej audycji "Halo komputer" próbowano popularyzować LOGO, iecz bez kontaktu z odbiorcą kursu trudno mówić o efektach. W TV m.in. wyemitowano angielski translator LOGO na ZX Spectrum, emisję powtórzono w Rozgłośni Harcerskiej. LOGO w tej postaci jest niestety mało zrozumiałe dla młodzieży i dzieci interesujących się programowaniem komputerów, ale nie znających angielskiego.

Pierwszy kontakt z komputerem jest niezwykle istotny dla utrwalenia nawyków na przyszłość. Powinien on być NATURALNY (a więc po polsku), ŁATWY (przyjazny dla robiących błędy), CIEKAWY (wciągający sceptyków) i SATYSFAKCJONUJĄCY (szybkie efekty nauki). Takie spotkanie może zapewnić LOGO w swej polskiej, zrozumiałej dla początkującego, nie najeżonej obcymi słowami wersji (nawet znający angielski myślą w ojczystym języku).

Dla opracowania polskiej wersji LOGO powstał pod egidą Polskiego Towarzystwa Informatycznego zespół, składający się m. in. z autorów wcześniejszych amatorskich propozycji, który uzgodnił standard polskiego LOGO. Pierwsza jego wersja prezentowana była w grudniowym numerze „Bajtka" i wzbudziła spore zainteresowanie, m.in. otrzymaliśmy wiele cennych uwag i propozycji. Po ich analizie i konsultacjach z zainteresowanymi środowiskami (m.in. na kursach nauczycielskich) oraz zebraniu doświadczeń dydaktycznych zakończono prace nad językiem i pierwszym translatorem polskiego LOGO na ZX Spectrum.

JĘZYK

Pierwszy krok polegał na ustaleniu polskiego brzmienia słów kluczowych, czyli na stworzeniu standardu języka (przy pozostawieniu możliwie nienaruszonej filozofii języka).

Oprócz nazw komend i operacji spolszczono także komunikaty o błędach i komentarze. Zbiór znaków wprowadzanych z klawiatury i pojawiających się na ekranie rozszerzono o litery charakterystyczne dla języka polskiego.

Obecnie najpilniejszym zadaniem jest napisanie i wydanie związanych z polskim LOGO podręczników i wydawnictw popularyzujących oraz opracowanie implementacji polskiego LOGO na wszystkie dostępne w kraju komputery. Prace te już podjęto.

TRANSLATOR

Równolegle z pracami teoretycznymi opracowano (Andrzej Bączyński i Zbigniew Kasprzycki) interpreter polskiego LOGO na ZX Spectrum. Jest on praktyczną realizacją propozycji zespołu i pierwszą kompletną wersją polskiego LOGO. Zarówno w nagłówku, jak i wewnątrz kodu, jest on sygnowany przez PTI i dalej określany będzie jako PTI LOGO. Wszelkie inne krążące po kraju wersje polskiego LOGO, w tym także nielegalnie skopiowane wcześniejsze projekty autorstwa członków zespołu, są w tej sytuacji jedynie atrapami. Jedynym wyjątkiem jest opracowana przez Tadeusza Wilczka nakładka na Sinclair LOGO SOLI-LCSI pt. INTERLOGO, która może być użyteczna jako narzędzie do automatycznego tłumaczenia programów z wersji angielskiej na polską i na odwrót. PTI LOGO będzie wkrótce rozprowadzane przez PTI dla szkół oraz przez państwowych dystrybutorów dla użytkow-

POLSKIE LOGO

Polskie LOGO opracowali: ANDRZEJ BĄCZYŃSKI, JAROSŁAW KANIA, ZBIGNIEW KASPRZYCKI, WŁADYSŁAW MAJEWSKI, PIOTR TYMOCHOWICZ, STANISŁAW WALIGÓRSKI I TADEUSZ WILCZEK.

POLSKIE LOGO

Po angielsku	Po polsku	
AND	I	spójnik logiczny
ARCCOS	ARCCOS	funkcja tryg.
ARCCOTANGENT ARCCOT	ARCCTG	funkcja tryg.
ARCSIN	ARCSIN	funkcja tryg.
ARCTANGENT ARCTAN	ARCTG	funkcja tryg.
ASCII	ASCII	kod ASCII znaku n
BACK, BK	WSTECZ, WS	żółw o n kroków wstecz
BACKGROUND, BG	TŁO^	zwraca kolor tła
BRIGHT	JASKRAWO	tło tekstu rozjaśnione
BUTFIRST, BF	BEZPIERW	lista BEZPIERWszego
BUTLAST, BL	BEZOST	lista BEZOSTatniego
BYE	DOŚĆ	powrót do BASICU
CATALOG	-------	
CHAR	ZNAK	znak o danym kodzie ASCII
CLEAN	ZMAŻ	czyści ekran, nie rusza żółwia
CLEARSCREEN CS	CZYŚĆ, CS	czyści ekran, żółw "do domu"
CLEARTEXT, CT	ZMAŻTEKST, ZT	usuwa teksty
COPYDEF	POWIEL	nowa nazwa procedury
COPYSCREEN	DRUKOBRAZU	ekran na drukarkę
COSIN, COS	COS	funkcja tryg.
COTANGENT. COT	CTG	funkcja tryg.
COUNT	DŁUGOŚĆ	liczba elementów listy
CURSOR	KURSOR^	zwraca położenie kursora (tekst)
DEFINE	OKREŚL	zdefiniuj procedurę (danej treści)
DEFINDP	OKREŚL?	czy jest procedura o danej nazwie
DIV	ILORAZ	działanie arytm.
DOT	PKT	postaw punkt
EDIT, ED	RED	przejście do REDagowania
EDNS	REDWN	redagowanie wszystkich nazw
EMPTYP	PUSTE?	czy lista (słowo) n jest pusta?
END	JUŻ	koniec definiowania procedury
EQUALP	RÓWNE?	równoznaczne z "=" (przedrostkowo)
ERASE, ER	USUŃ, US	usuń procedurę z pamięci
ERALL	USW	usuń wszystko z pamięci
ERASEFILE	-------	
ERN	USN	usuń nazwę ("zmiennej")
ERNS	USWN	usuń wszystkie nazwy ("zmiennych")
ERPS	USWP	usuń wszystkie procedury
FALSE	FAŁSZ	wartość logiczna
FENCE	POLE	ekran graficzny to zamknięte pole
FIRST	PIERW	PIERWszy element listy (operacja)
FLASH	MIGAJ	wyświetlany tekst będzie "migał"
FORWARD, FD	NAPRZÓD, NP	żółw o daną liczbę kroków naprzód
FPUT	NAP	(NA Początek) operacja na liście
HEADING	KĄT^	zwraca kierunek (kąt) żółwia
HIDETURTLE, HT	SŻ	schowaj żółwia (żółw niewidoczny)
HOME	WRÓĆ	żółw wraca do środka ekranu
IF	JEŚLI	instrukcja (operacja!) warunkowa
INT	ENTIER, ENT	część całkowita liczby
INVERSE	NEGATYW	wyświetlany tekst w negatywie
ITEM	ELEMENT	zwraca n-ty element listy
KEYP	KLAWISZ?	sprawdza stan klawiatury
LAST	OST	OSTatni element listy
LEFT, LT	LEWO, LW	żółw w lewo o daną liczbę stopni
LIST	LISTA	operacja tworzenia listy z
LISTP	LISTA?	czy dany obiekt to lista?
LOAD	ŁADUJ	wczytywanie z magnetofonu
LOADD	ŁADUJR	wczytywanie zawartości edytora
LOADSCR	ŁADUJO	wczytywanie ekranu
LPUT	NAK	(NA Koniec) operacja na liście
MAKE	PRZYPISZ, PRZYP	instrukcja (komenda) przypisania nazwy do obiektu
MEMBERP	ELEMENT?	czy dany obiekt jest na liście
NAME	NAZWIJ	instrukcja nazwania obiektu
NAMEP	JEST?	czy to nazwa jakiegoś obiektu
NODES	WOLNE	liczba wolnych jednostek pamięci
NORMAL	POZYTYW	odwrotność negatywu
NOT	NIE	operacja logiczna
NUMBERP	LICZBA?	czy dany obiekt jest liczbą
OR	LUB	operacja logiczna
OUTPUT, OP	WYNIK, WY	przypisanie wartości procedurze
OVER	NADRUK	pisz "jedno na drugim"
PENCOLOUR, PC	PISAK	zwraca kolor pisaka
PENDOWN, PD	OPU	OPUść pisak (rysuj)
PENERASE, PE	ŚCIERANIE	odwróć pisak gumką w dół (ścieraj)
PENREVERSE, PX	ODWRACANIE	rysuj lub ścieraj (jeśli kreska)
PENUP, PU	POD	PODnieś pisak (nie rysuj)
PO	PO	POkaż treść danych procedur
POALL	POW	POkaż Wszystko
PONS	POWN	POkaż Wszystkie Nazwy

POPS	POWP	POkaż Wszystkie Procedury
POSITION, POS	POZˆ	zwraca pozycję żółwia
POTS	POTP	POkaż Tytuły Procedur
PRIMITIVEP	PIERWOTNE?	czy to słowo jest pierwotne?
PRINT, PR	PISZ, PP	pisz na ekran (tekstowo)
PRINTOFF	WYŁĄCZDRUK	wyłącz drukarkę
PRINTON	WŁĄCZDRUK	włącz drukarkę
PRODUCT	ILOCZYN	działanie arytmetyczne
RANDOM	LOSOWA	zwraca losową liczbę (od 0 do n-1)
READCHAR, RC	CZYTAJZNAK, CZ	zwraca znak wciśniętego klawisza
READLIST, RL	CZYTAJLISTĘ, CL	zwraca listę wprowadzoną z klawiatury (tzn. linię do ENTER)
RECYCLE	ODŚMIEĆ	uporządkuj pamięć
REMAINDER	RESZTA	zwraca resztę z dzielenia
REPEAT	POWTÓRZ	pętla n-krotna (POWTÓRZ n razy)
RIGHT, RT	PRAWO, PW	żółw w prawo o daną liczbę stopni
ROUND	ZAOKR	zaokrąglij liczbę (do całkowitej)
RUN	ZRÓB	wykonaj instrukcje z listy
SAVE	ZAPISZP	zapisz procedury (dane) na taśmę
SAVEALL	ZAPISZ	zapisz wszystko na taśmę
SAVED	ZAPISZR	zapisz zawartość edytora na taśmę
SAVESCR	ZAPISZO	zapisz zawartość ekranu na taśmę
SCRUNCH	PROPORCJAˆ	zwraca proporcje ekranu
SENTENCE, SE	ZDANIE, ZD	operacja tworzenia listy
SETBG	TŁO	ustawia kolor tła na dany
SETBORDER, SETBR	RAMKA	ustawia kolor ramki (marginesu) na dany
SETCURSOR, SETCUR	KURSOR	ustawia kursor (tekstowy) na danej pozycji
SETDRIVE	----	
SETHEADING, SETH	KĄT	żółw obraca się na dany kierunek (na dany kąt)
SETPC	PISAK	zmień kolor pisaka na dany
SETPOS	POZ	żółw na daną współrzędną
SETSCR	PROPORCJA	ustawia proporcje ekranu
SETTC	KOLORYT	ustawia kolor tekstu i tła
SETX	XPOZ	żółw w poziomie do danej pozycji
SETY	YPOZ	żółw w pionie do danej pozycji
SHOW	POKAŻ	pokaż typ obiektu (wyświetl)
SHOWNP	WIDAĆ?	czy żółw jest widoczny?
SHOWTURTLE, ST	PŻ	Pokaż Żółwia
SINE, SIN	SIN	funkcja tryg.
SOUND	NUTA	wydaj dźwięk
SQRT	PIERWIASTEK, PKW	funkcja arytmetyczna
STOP	STOP	przerwij działanie procedury
SUM	SUMA	działanie arytmetyczne
TANGENT, TAN	TG	funkcja tryg.
TEXT	TREŚĆ	zwraca treść procedury (operacja)
TEXTCOLOUR, TC	KOLORYTˆ	zwraca kolor tekstu i tła
TEXTSCREEN, TS	TEKSTY, TS	zmień ekran na tekstowy
THING	WARTOŚĆ, WAR	wartość, jaką reprezentuje nazwa
TO	OTO	początek definicji procedury
TOPLEVEL	PRZERWIJ	przerwij wszystkie obliczenia
TOWARDS	AZYMUT	zwraca kierunek na dany punkt
TRUE	PRAWDA	wartość logiczna
TYPE	WPISZ	pisz w jednej linii
WAIT	CZEKAJ	czekaj daną liczbę kwantów czasu
WINDOW	OKNO	ekran to okienko w przestrzeni
WORD	SŁOWO	zrób słowo z danych elementów
WORDP	SŁOWO?	czy dany obiekt to słowo?
WRAP	SKLEJ	sklej ekran graficzny w torus
XCOR	XPOZˆ	zwraca współrzędną X żółwia
YCOR	YPOZˆ	zwraca współrzędną Y żółwia
------------	LN	logarytm – funkcja arytmetyczna
------------	EXP	"e do x" – funkcja arytmetyczna
------------	PIˆ	zwraca wartość stałej "pi"
------------	MIGI	migająca grafika żółwia
------------	MIGO	powrót do zwykłej grafiki żółwia
------------	JASI	rozjaśniona grafika żółwia
------------	JASO	powrót do zwykłej grafiki żółwia
------------	ZAMALUJ, ZAM	zamaluj figurę, w której jest żółw
.BLOAD	.ŁADUJBIN	wczytaj bajty z magnetofonu
.BSAVE	.ZAPISZBIN	zapisz bajty na taśmę
.CALL	.WYWOŁAJ	wywołaj program w asemblerze
.CONTENTS	.PAMIĘĆ	co LOGO pamięta (jakie słowa)?
.DEPOSIT	.UMIEŚĆ	wpisz liczbę do komórki pamięci
.EXAMINE	.ZOBACZ	odczytaj zawartość komórki pamięci
.PRIMITIVES	.PIERWOTNE	wyświetl słowa pierwotne LOGO
.RESERVE	.ZAJMIJ	zajmij (zarezerwuj) pamięć
.RESERVED	.ZAJĘTE	zwraca adres i długość pamięci
.SERIALIN	.PRZYJMIJ	przyjmij bajt z portu (RS232)
.SERIALOUT	.WYŚLIJ	wyślij bajt do portu (RS232)
.SETSERIAL	.TRANSMISJA	ustaw prędkość transmisji (dysk)
----------	.ZOBACZP	odpowiednik BASICowego OUT
----------	.UMIEŚĆP	odpowiednik BASICowego IN

SPIS KOMUNIKATÓW SYSTEMU

"Not enough inputs to ..." — "Za mało danych dla ..."
"I don't know how to ..." — "Nie wiem jak zrobić ..."
"You don't say what to do with ..." — "Nie powiedziałeś co zrobić z ..."
"... does not output to ..." — "... nie daje wyniku potrzebnego ..."
"... is already defined ..." — "... jest już zdefiniowane"
"... is not true or false" — "... nie jest ani prawdą, ani fałszem"
"... defined" — "... zdefiniowane"
"To many inside parentheses" — "Za dużo wewnątrz nawiasów"
"File not found" — "Nie ma takiego pliku"
"Bad file name" — "Zła nazwa pliku"
"You are at toplevel" — "Jesteś na najwyższym poziomie"
"STOPPED!!!" — "PRZERWANE!!!"
"Turtle out of field" – "Żółw poza polem"
"Not enough space to proceed" — "Za mało pamięci by kontynuować"
"... doesn't like ..." — "... nie chce ..."
"... as input" — "... jako danej"
"... has no value" — "... nie ma wartości"
"... in ..." — "... w ..."
"... is a primitive" — "... jest specjalnym słowem"
"Not enough items in ..." — "Za mało elementów w ..."
"Overflow" — "Przepełnienie"
"Can't divide by zero" — „Nie można dzielić przez zero"
"Number to big" — „Za duża liczba"
"ZX Interface I not present" — "To niestety nie działa"
"LOGO Editor ..." — "Edytor LOGO ..."
"Start tape, then press any key" — "Włącz magnetofon i wciśnij coś"
"Tape loading error" — "Błąd taśmy"

Przedostatni komunikat nie jest może zbyt elegancki, ale z przyczyn obiektywnych (musi się mieścić w jednej linii na ekranie!) trudno go poprawić.

ników prywatnych, ostrzegamy więc przed nabywaniem go u handlarzy i przed próbami sprzedawania bez zgody autorów.

Interpreter jest globalną przeróbką Sinclair LOGO firmy SOLI-LCSI, a więc lwia część procedur obsługujących analizę syntaktyczną i interpretację kodu pośredniego jest zaczerpnięta lub nawet "żywcem" przepisana z kodu interpretera angielskiego. Istotną zmianą są polskie litery, wprowadzane po uprzednim wciśnięciu klawisza "GRAPH", oczywiście polskie słowa i komunikaty oraz nowe komendy i operacje. Zmiany te nie są drobne, gdyż polskie teksty i słowa są z reguły dłuższe od angielskich, a fragmenty translatora są nierelokowalne. Wprowadzenie polskich liter wymagało stworzenia pakietu procedur obsługujących wejście/wyjście – diametralnie różniących się od stosowanych w Sinclair LOGO. Przy okazji udało nam się poprawić kilka błędów firmy SOLI-LCSI.

POPRAWKI

Dla przykładu w PTI LOGO pamięć nie zapycha się nieodwracalnie i nie zastyga ono z pytaniem "scroll" po użyciu instrukcji KURSOR (SETCUR) w trybie graficznym. Odziedziczyło ono jednak po Sinclair LOGO kilka poważnych błędów, których lista jest nadal otwarta. Dlatego pracy nad interpreterem nie przerywamy i stopniowo upowszechniać będziemy jego kolejne, coraz doskonalsze wersje. Obecna jest pierwszą na tyle dojrzałą, by mogła być prezentowana publicznie.

W PTI LOGO zrezygnowaliśmy z komend dotyczących dyskietek i microdrive'ów. Realnie patrząc mało kto współpracuje w LOGO na ZX Spectrum z tymi nośnikami pamięci.

PTI LOGO zajmuje 26000 bajtów pamięci i zaczyna się od komórki o adresie 24020. Na struktury użytkownika przeznaczonych jest więc ok. 15000 komórek pamięci. W stosunku do Sinclair LOGO jest dłuższy o ok. 800 bajtów, ale kosztem nie struktur użytkownika, lecz BASICU. W efekcie po przejściu do BASICU (tylko po co?) możemy wykonywać tylko pojedyncze rozkazy (i to dość krótkie), by ich treścią nie zamazać interpretera LOGO.

Interpreter uruchamiamy (restart) przez RANDOMIZE USR 24832 lub (jeśli nie zmieniliśmy programu obsługującego w BASICU) przez GO TO 3, a wznawiamy jego działanie bez niszczenia struktur w LOGO przez RANDOMIZE USR 24835 lub GO TO 2.

UWAGA! PTI LOGO nie może działać, gdy do ZX Spectrum podłączony jest ZX Interface 1.

➧ 12

11 ◆

NOWE OPERACJE I KOMENDY

PTI LOGO wyposażone jest w dodatkowe operacje i komendy, nie istniejące w wersji angielskiej. Są to operacje arytmetyczne LN, EXP, PI ↑ znaczące po kolei: logarytm naturalny, "e do x" oraz stała pi. Jest także 6 nowych komend graficznych. Ich kombinacje dają ciekawe i zaskakujące efekty. Są to:

JAS1	żółw porusza się po rozjaśnionym tle. Po wykonaniu CS jasny jest cały ekran.
JAS0	wyłączenie JAS1.
MIG1	żółw na tle migającym. Po CS miga cały ekran.
MIG0	wyłączenie MIG1 (cały ekran przestaje migać po CS)
NEG	negatyw ekranu, także tekstowego, nie mylić z komendą NEGATYW odnoszącą się do tekstów pisanych. Ciekawy efekt daje wywołanie procedury: OTO MIG / NEG MIG / JUŻ
ZAMALUJ, ZAM	wypełnia kolorem żółwia obszar zamknięty, wewnątrz którego znalazł się żółw.

Ta ostatnia komenda jest najbardziej atrakcyjna. Pozwala ona dowolnie kolorować uprzednio narysowane figury, co stwarza jakościowo nowe możliwości graficzne. W Sinclair LOGO efektywne kolorowanie konturów było praktycznie niemożliwe.

Dobrze jest znać strategię zamalowywania figur, bo zdarzyć się może, że nagle zostanie nam niepokolorowane miejsce. Otóż z miejsca gdzie jest żółw (może być widoczny lub nie) LOGO prowadzi pionową kreskę w górę i w dół, aż do napotkania brzegu figury (lub ekranu). Z każdego punktu owej kreski prowadzi teraz kreski poziome (w lewo i w prawo), aż do napotkania brzegów figury. Wynika z tego, że aby zamalować koło, żółw musi znaleźć się w pobliżu pionowej osi symetrii tego koła. W innym przypadku zdarzyć się może, że zostanie zamalowany tylko poziomy pas w kole, bo pionowa kreska z pozycji żółwia natrafi na brzegi koła leżącego poniżej jego najwyższego punktu i powyżej najniższego. O tej strategii zamalowywania warto więc wiedzieć, by oszczędzić sobie niespodzianek i straty czasu na jałową pracę. Dobrze jest też podnieść pisak żółwia (POD) przed wejściem nim do środka figury, gdyż kreska rysowana przez niego wewnątrz figury może mieć znaczenie przy kolorowaniu. Poniżej przedstawiam procedurę rysującą kolorowe koło, wykorzystującą komendę ZAMALUJ (ZAM)

```
OTO KOŁO :R :KOLOR
PISAK :KOLOR OPU
POWTÓRZ 36 [NAPRZÓD 2*PI ↑ *:R/36 PRAWO 10]
POD PRAWO 90 NAPRZÓD :R ZAMALUJ WSTECZ :R LEWO 90
JUŻ
```

Zauważmy, jak łatwo otrzymać obrazek tak często drukowany na wszelkich reklamach komputerów – kolorowy wykres w postaci słupków, obrazujących zjawiska ekonomiczne itp. Wystarczy rysować prostokąty o odpowiedniej wysokości i je kolorować. Należy jednak pamiętać, że ekran na ZX Spectrum (także graficzny) jest podzielony na kwadraciki 8 punktów na 8 i w obrębie każdego kwadracika jest dostępny tylko jeden kolor. Trzeba zatem tak dobierać sąsiedztwo kolorów, by nie "włazily" one na siebie (jeśli nam na tym zależy).

Na poprzednich stronach przedstawiony jest ostateczny spis słów kluczowych polskiego LOGO (PTI LOGO) oraz spis komunikatów.

JAK PRZYSPIESZYĆ DZIAŁANIE PROGRAMU?

Nawet dobrze napisany program może nie spełnić stawianych przed nim celów, gdy "liczy" zbyt szybko lub zbyt wolno. Tempo realizacji programu dydaktycznego musi być dostosowane do możliwości odbioru przez uczniów. Zwolnienie tempa pracy programu nie przedstawia specjalnych trudności nawet dla początkującego programisty. Służy do tego np. dyrektywa PAUSE m. Dużo trudniej przyspieszyć działanie programu.

Można tego dokonać drogą:

– zmiany algorytmu, według którego rozwiązano problem

– poprawienia programu napisanego w języku BASIC

– kompilacji programu napisanego w języku BASIC.

Jeżeli metody te okazały się niewystarczające, to można jeszcze:

– napisć program w innym języku wyższego poziomu (na przykład w języku Pascal w przypadku mikrokomputera ZX Spectrum)

– napisać program w języku mikroprocesora lub podobnym.

Ostatnia z podanych metod jest bardzo trudna, a w przypadku próby napisania całego programu na przykład w asemblerze mikroprocesora Z80 – bardzo czasochłonna. Często wystarczy jednak wykorzystanie gotowych procedur napisanych w języku maszynowym.

My skoncentrujemy się na trzech pierwszych metodach.

Zmiana algorytmu i przeredagowanie programu

Jest to najprostszy sposób przyspieszenia pracy programu, lecz należy przestrzec przed nadużywaniem tej właśnie metody. Przejrzysty algorytm może być bardzo potrzebny przy wszelakich próbach jego modyfikacji. Nadużywanie różnych trików powoduje, że program staje się nieczytelny nawet dla autora. Wszelakie poprawki należy stosować więc jedynie wtedy, gdy poprawiony algorytm będzie nie gorzej czytelny niż pierwotny. Istnieją pewne zasady dotyczące układania algorytmów jak i kodowania, przyspieszające czas realizacji programu. Poniżej zostanie podanych jedynie pięć najważniejszych zasad:

1. Należy unikać wielokrotnego, a niepotrzebnego wykonywania tych samych operacji. Na przykład raz obliczona wartość powinna być zapamiętana.

2. Obliczenia wymagają odpowiedniego redagowania. Należy pamiętać, że mnożenie jest wykonywane wielokrotnie szybciej niż potęgowanie czy dzielenie. Na przykład instrukcja LET a=b/2 jest wykonywana 1.5 razy dłużej niż LET a=bx0.5, podobnie instrukcja LET d=c 2 jest liczona 2.7 razy dłużej niż LET d=c x c. (Pierwsze dwie zasady nabierają szczególnego znaczenia, gdy znajdują się wewnątrz pętli i są wykonywane wielokrotnie).

3. Należy usuwać poza pętle operacje, które mogą być wykonane wcześniej lub później. Należy rezygnować z nieistotnych pętli. Przykładowo pętla FOR I=1 TO 5: LET a(I)=I: NEXT I to tylko dwie instrukcje mniej niż pięć instrukcji podstawienia (a(1)=1 ... a(5)=5), za to czas realizacji pętli jest trzykrotnie dłuższy.

4. Kilka pętli, o ile jest to możliwe, należy łączyć w jedną.

Zamiast FOR I=1 TO m: LET a(i)=i :NEXT i :FOR J=1 TO m: LET b(J)=m-J : NEXT J wystarczy jedna pętla FOR I=1 TO m : LET a(I)=I : LET b(I)=m-I :NEXT I.

5. Zagłębienie pętli (tzn. umieszczanie pętli wewnątrz innej) należy organizować tak, by pętle wewnętrzne były wykonywane częściej niż zewnętrzne. Daje to oszczędności czasu przy otwarciach i zamknięciach pętli. Dla trzech pętli FOR k=1 TO 40 : FOR I=1 TO 20 : FOR m=1 TO 10 optymalny sposób zapisu to FOR m=1 TO 10 : FOR I=1 TO 20 : FOR k=1 TO 40.

Jak widać z podanych uwag, "wąskim gardłem" są wszelkie pętle i dlatego organizacja i optymalizacja pętli może przynieść dość duże oszczędności czasu. Często jednak efekty te są nie zadowalające.

Kompilacja programu

Jeżeli dotychczas działania nie przyniosły spodziewanych efektów, to gotowy i poprawnie napisany program należy skompilować. Co to znaczy?

Większość sprzedawanych mikrokomputerów wyposażona jest w interpreter języka BASIC zapisany w pamięci typu ODCZYT (ROM). Zaletą interpretera jest to, że każda wprowadzona instrukcja nie posiadająca numeru linii może być przetłumaczona na język mikroprocesora (zinterpretowana) i od razu wykonana. Wadą jest to, że w ten sam sposób wykonywane są instrukcje posiadające swoje numery. Oznacza to, że wielokrotne wykonanie instrukcji wymaga wielokrotnego tłumaczenia jej na kod maszynowy.

Kompilator to program tłumaczący program napisany w jednym języku na program w kodzie akceptowanym przez mikroprocesor. Program taki wprowadzany jest zwykle z urządzenia zewnętrznego (na przykład z kasety czy dyskietki). Po przetłumaczeniu realizowany jest program zapisany w kodzie maszynowym. Odpada więc wielokrotne tłumaczenie tej samej instrukcji. Dobry kompilator nie jest oprogramowaniem tanim. Istnieje kilka kompilatorów języka BASIC mikrokomputera ZX Spectrum, które można nabyć. Każdy z nich ma swoje zalety, i niestety wady. Dostępne kompilatory można podzielić na kompilatory całkowitoliczbowe i kompilatory pełne. Omówimy tu cztery z nich: IS oraz FP firmy Softek, MCODER II firmy PSS i BLAST firmy Oxford Computer Systems spośród komputerów pełnych.

Kompilatory IS i FP firmy Softek

Oba kompilatory dostępne są w wersjach 16 KB i 48 KB dla odpowiedniego modelu mikrokomputera ZX Spectrum. Ponieważ grono użyt-

kowników kompilatorów 16 KB będzie niewielkie, wszystkie uwagi dotyczą wersji 48 KB (dokładniej wersji 48 KB 1.7). Jak wspomniano, kompilator IS jest kompilatorem całkowitoliczbowym o możliwościach przyspieszenia pracy nawet dwustukrotnie, zaś FP to kompilator pełny, przyspieszający prace programu nawet dwudziestokrotnie. W naszych testach otrzymano pełne potwierdzenie przyspieszenia pracy przez kompilator IS. W przypadku kompilatora FP uzyskano jedynie pięciokrotne przyspieszenie realizacji programu.

Kompilator ładujemy do pamięci mikrokomputera tak jak program napisany w języku BASIC instrukcją LOAD "". Na kasecie znajdują się trzy bloki – program ładujący i obsługi ekranu oraz dwa bloki kodu maszynowego. Zasadnicza część kompilatora liczy 6100 bajtów i zajmuje obszar powyżej adresu 59 300. Obszar pamięci od adresu 40 000 jest obszarem roboczym rezerwowanym na kod wynikowy kompilatora. Jest to obszar bardzo duży. Można go zmniejszyć (lub zwiększyć) instrukcją CLEAR n-1, gdzie n oznacza adres pierwszej komórki pamięci, do której chcemy załadować kod wynikowy. Pozwala to na przechowanie w pamięci mikrokomputera jednocześnie kilku skompilowanych programów (patrz przykład w pkt. 5.1.7).

Po załadowaniu kompilatora wciśnięcie dowolnego klawisza powoduje wymazanie programu ładującego i użytkownik mający do dyspozycji obszar pamięci do adresu 39999 (lub n-1) nie odczuwa obecności kompilatora w pamięci mikrokomputera. Pozwala to na załadowanie programu z kasety lub wprowadzenie tekstu programu bezpośrednio z klawiatury. Program powinien być przetestowany z wykorzystaniem dyrektywy RUN lub odpowiedniej. Dopiero dobrze napisany program, którego wyniki znane są użytkownikowi, może być poddany kompilacji.

Kompilację rozpoczynamy instrukcją RANDOMIZE USR 59 300. Przebieg kompilacji możemy śledzić na ekranie monitora. W przypadku wystąpienia błędu kompilacja zostaje przerwana ze wskazaniem niezrozumiałego miejsca w programie. Błędy mogą pojawić się w trakcie kompilacji, mimo iż program pracował poprawnie. Wynika to z ograniczeń kompilatora (na przykład IS nie pozwala na użycie tablic). Zlokalizowany przez kompilator "błąd" pozwala na modyfikacje programu i wykonanie ponowne kompilacji.

Poprawna kompilacja kończy się informacją o braku błędów, podaniem początkowego i końcowego adresu zajmowanego przez skompilowany program oraz końcowego adresu obszaru zmiennych. Na przykład:

START ADDRES 40 000
END ADDRES 40 424
VARIABLE END 40 445

Skompilowany program nazywać będziemy dalej programem w kodzie maszynowym.

Program w kodzie maszynowym można uruchomić za pomocą dyrektywy RANDOMIZE USR 40 000 lub RANDOMIZE USR n (o ile wykonano CLEAR n – 1). Należy sprawdzić czy efekt działania programu w BASICU i kodzie maszynowym są identyczne. Może się zdarzyć (o czym mieliśmy okazję się przekonać), że rezultaty są różne. Wynika to z tego, że każdy kompilator posiada swoje ograniczenia, a nie wszystkie z nich są publikowane w instrukcjach obsługi. Różnice te wynikają ze struktury programu. Nie stwierdzono, by program pracujący dla jednego zestawu danych nie działał dla innego.

Każdorazowe wykonanie programu wymaga obecności kompilatora w pamięci mikrokomputera, ponieważ program ten w trakcie realizacji odwołuje się do procedur bibliotecznych. Fakt ten narzuca następujące możliwości składowania programu:

(n jest równe 40 000 lub użyte w instrukcji CLEAR n-1)
– SAVE "nazwa" CODE n, 65536-n
(Program taki można załadować instrukcją LOAD "" CODE i uruchomić RANDOMIZE USR n)
– SAVE "nazwa" CODE n, długość, gdzie długość obliczono na podstawie znajomości adresu końcowego i początkowego obszaru zajmowanego przez program (patrz pkt. 5.1.4)
(Program taki należy ładować i uruchamiać po uprzednim załadowaniu kompilatora. Metoda ta oszczędza czas ładowania i jest zalecana, gdy chcemy korzystać z kilku skompilowanych programów).

Załóżmy, że w trakcie zajęć chcielibyśmy wykorzystać dwa krótkie programy POKAZ i TEST w wersji skompilowanej. Oba programy wymagają kompilatora pełnego. Programy są nagrane na kasetę. Tryb przygotowania programów do zajęć będzie następujący:

– ładujemy kompilator LOAD""
– wciskamy dowolny klawisz i zmieniamy RAMTOP – CLEAR 49999
– ładujemy pierwszy program – LOAD "POKAZ"
– kompilujemy pierwszy program – RANDOMIZE USR 59300
– notujemy ostatni adres komórki programu (np. 56453)
– testujemy program w kodzie maszynowym – RANDOMIZE USR 50000
– ponownie zmieniamy RAMTOP – CLEAR 39999
– ładujemy drugi program – LOAD "TEST"
– kompilujemy drugi program – RANDOMIZE USR 59300
– notujemy ostatni adres komórki programu (np. 48732)
– testujemy drugi program – RANDOMIZE USR 40000
– nagrywamy na kasetę programy i kompilator – SAVE "PROGRAMY" CODE 40000, 25536 (lub tylko program SAVE "Pokaz" CODE 50000, 6454: SAVE "test" CODE 50000, 8733)

Niezależnie od metody nagrywania w trakcie zajęć uruchamiamy program TEST – RANDOMIZE USR 40000, a program POKAZ – RANDOMIZE USR 50000.

Jeżeli skompilowany program nie posiada instrukcji STOP, to może być wykorzystany jako podprogram w kodzie maszynowym przez inny program napisany w języku BASIC. Jest to szczególnie istotna zaleta kompilatorów IS i FP, w przypadku gdy nie możemy skompilować całego programu.

KOMPILATOR MCODER II FIRMY P.S.S.

Kompilator MCODER II jest bardzo podobny do kompilatora IS i może być wykorzystywany zamiennie. Dodatkową jego zaletą jest lepsza instrukcja, dzięki której większość ograniczeń jest znana. Istnieje możliwość definiowania tablic, ale tylko jednowymiarowych. Wszystkie podane uprzednio uwagi można stosować do kompilatora MCODER II pamiętając, że do uruchomienia kompilacji służy instrukcja RANDOMIZE USR 60000.

KOMPILATOR BLAST FIRMY O.C.S.

Kompilator ten daje lepsze rezultaty od prezentowanego kompilatora FP. Pozwala na kompilacje prawie każdego programu realizowanego na mikrokomputerze ZX Spectrum. Niestety możliwość ta dotyczy wersji 4.0, podczas gdy w kraju (stan na 1985-12-31) dostępna jest jedynie bardzo niedopracowana wersja 1.1. Kompilator ten przeznaczony jest dla posiadaczy szybkiej pamięci kasetowej (MICRODRIVE) lub krótkich programów, gdyż po załadowaniu pozostawia jedynie 2689 wolnych komórek pamięci. W przypadku dłuższych programów jest możliwa praca z wykorzystaniem magnetofonu kasetowego i dwóch kaset. Praca ta jest niestety uciążliwa i nie zalecana początkującym użytkownikom mikrokomputera. Program do kompilacji należy przygotować z wykorzystaniem specjalnego dostarczanego osobno programu narzędziowego TOOLKIT. W trakcie kompilacji taśmy z programem do czytania i czyste do zapisu wykorzystywane są na przemian.

Kompilator BLAST daje dwie możliwości kodu wynikowego:

– kod mikroprocesora Z80 zwany przez nas kodem maszynowym
– P-KOD firmy BLAST, który jest tajemnicą firmy. Wiadomo jedynie, że program wynikowy zajmuje mniej miejsca niż w języku BASIC i że liczy się wolniej niż w kodzie maszynowym.

Kompilator można załadować jak każdy program napisany w języku BASIC instrukcją LOAD"". Ładowanie jest bardzo długie i trwa ponad pięć minut. Po załadowaniu kompilator żąda podania haseł uniemożliwiając w ten sposób dostęp do programu osobom niepowołanym. Tylko poprawne odpowiedzi na wszystkie pytania umożliwiają dalszą pracę.

Specjalne dyrektywy umieszczone w instrukcjach REM programu napisanego w języku BASIC sterują przebiegiem kompilacji. O tym że instrukcja REM jest dyrektywą nie komentarzem decyduje znak! jako pierwszy po REM. Istotne są następujące dyrektywy:

REM! MACHINE CODE – wynik działania BLASTA to kod maszynowy
REM! PCODE – daje P-KOD
REM! AUTORUN – automatyczne uruchomienie programu po załadowaniu
REM! INT lista – lista zawiera nazwy zmiennych całkowitych, co przyspiesza działanie programu

Jeżeli w programie nie umieścimy żadnej z dwóch pierwszych dyrektyw, to program nasz będzie tłumaczony na P-KOD.

Komendy poprzedzone znakiem mnożenia – x traktowane są jako instrukcje kompilatora. Istotne są następujące instrukcje:

XC – kompilacja programu
XR – uruchomienie skompilowanego programu
XN – "wyczyszczenie" obszaru ubocznego kompilatora
XS – nagranie programu wynikowego na kasetę
XI – określenie skąd ma być czytany program (kaseta, pamięć RAM czy pamięć szybka – MICRODRIVE)
XO – określenie gdzie ma być składowany program (możliwości jak wyżej).

Jeżeli nie użyjemy instrukcji XI i XO, to program będzie czytany z pamięci (trzeba go tam wcześniej załadować np. poprzez LOAD ""), również program wynikowy zostanie umieszczony w pamięci RAM mikrokomputera.

Zapisanie programu na kasecie odbywa się z wykorzystaniem instrukcji XS. Na pytanie kompilatora gdzie? należy wybrać kasetę, a następnie podać nazwę programu. Tak zapisany program ładujemy jak program napisany w języku BASIC instrukcją LOAD"" lub LOAD "nazwa", a nie jak program napisany w kodzie maszynowym.

TADEUSZ GOLONKA
ALINA PIETRZYK

OTO MERITUM

"MERITUM – Twój pierwszy krok w świat mikrokomputerów!". Hasło to rozbrzmiewa z Zabrza od połowy 1983 r.

Zaczęło się w Politechnice Śląskiej, gdzie na jednym roku studiowali m.in. obecny dyrektor techniczny Zakładów Urządzeń Komputerowych MERA-ELZAB Zygmunt Korga i obecny właściciel zakładu rzemieślniczego AGAPOL w Katowicach – Brynowie Paweł Podsiadło. Spotkali się ponownie po latach, by wspólnie zrobić coś, co byłoby ciekawą przygodą i powodem do chwały dla twórców, a zarazem dobrym interesem dla ich firm.

Hierarchia motywów była właśnie taka: własna satysfakcja z pracy, sława dla firmy, dopiero na koniec zyski. Słowo o sławie: dzięki produkcji inteligentnych terminali do dużych komputerów Elzab mieści się na liście 500 największych zakładów przemysłowych kraju oraz 100 największych eksporterów.

Poza macierzystym regionem firma, przynosząca krajowi wiele dewiz, pozostawała jednak praktycznie nie znana. Produkt prawie rynkowy, jakim jest masowy i popularny mikrokomputer, choć w najlepszym momencie nie stanowił więcej niż 3% wartości produkcji firmy, dawał szansę na nową tożsamość: pojawili się chętni do pracy, gazety pisały o śląskim producencie, dyrektorzy mieli towar atrakcyjny w krajowych przetargach, na Targach Poznańskich wokół stoiska Elzabu było tłoczno.

Paweł Podsiadło był w tym czasie pracownikiem przedsiębiorstwa zagranicznego ITM. Również dla tej firmy MERITUM stanowiło zawsze margines obrotów – i zarazem jedyny produkt, który przynosił jej ogólnopolski rozgłos. Trudno się więc dziwić, że szczególnie pieczołowicie twórcy wybierali nazwę dla swego komputerka. Jest ona majstersztykiem: zawiera w sobie zarówno trzon słowa MERA, jak i litery ITM.

* * *

Jako wzorzec dla MERITUM przyjęto amerykański komputer Tandy-Radio-Shack model II, będący ok. 1980 r. jednym z najpopularniejszych w USA. Wybór był trafny technicznie – TRS-80 łączył taniość i prostotę konstrukcji z możliwościami wykraczającymi poza typowy wzorzec mikrokomputera domowego, dopuszczał całkiem poważne zastosowania profesjonalne. Podobny wybrali zresztą Węgrzy, których najpopularniejszy szkolny mikrokomputer HT-1080Z oparty został na tym sam wzorcu. Wybór ten pociągał za sobą także pewne negatywne konsekwencje techniczne: TRS-80 był na rynku amerykańskim "ostatnim mohikaninem", ostatnim odnoszącym sukcesy przedstawicielem pierwszego pokolenia mikrokomputerów, które dla oszczędniejszego wykorzystania pamięci rezygnowały z pełnej rozdzielczości grafiki, stosując znakowy opis ekranu w pamięci.

Wada ta, aczkolwiek najczęściej podnoszona przez krytyków MERITUM, którzy często demagogicznie porównywali ten model z zabawkami w rodzaju ZX 81, nie ma charakteru podstawowego, lecz wtórny. W końcu IBM PC również nie dysponuje grafiką na głównej karcie, a wymienne karty graficzne każdy użytkownik dokupuje sobie za cenę rzędu 10-15% wartości urządzenia. Dla MERITUM również istnieje już dziś – o dwa lata za późno – procesor graficzny o możliwościach przewyższających ZX Spectrum w takim samym stopniu, z jakim ZX Spectrum wyprzedzało ZX 81.

Główną wadą wrodzoną MERITUM były warunki jego narodzin wraz z pozatechnicznymi wadami wyboru wzorca: Tandy-Radio-Shack nie jest firmą komputerową, lecz wielkim przedsiębiorstwem handlowym dysponującym ogólnoamerykańską siecią sklepów z elektroniką użytkową, nie czuła więc ani potrzeby, ani konieczności rozwijania swego modelu. Gdy minął jego czas, pod tą samą marką (model 1000) zaczęto zwyczajnie sprzedawać zupełnie od poprzednika różny produkt... koreański, którego dostawca zaoferował najlepsze warunki handlowe i zrezygnował z własnej marki. W chwili więc gdy MERITUM zaczynało być lansowane, o jego wzorcu w USA już zapomniano, znikała poświęcona mu literatura i rynek oprogramowania. Równocześnie zwolennicy MERITUM pobierali bolesną lekcję geografii. Oto nabywcy sprzętu Sinclaira, firmy znacznie mniejszej niż TRS i nawet na rynku mikrokomputerowym mającej mniejsze niż TRS obroty, pławili się dosłownie w zalewie płynących z W. Brytanii, Francji, a nawet Hiszpanii dóbr: oprogramowania, literatury, czasopism, informacji technicznej dla majsterkowiczów. Ameryka okazała się być na tyle oddalona, że przepływ podobnych materiałów dotyczących TRS-80 nigdy nie osiągnął liczącej się dla użytkowników MERITUM skali. Do dziś mogą oni liczyć tylko na własne siły, na własną przemyślność.

Od producenta, przez najcenniejsze pierwsze pół roku od dnia premiery, w czasie gdy pokłady ludzkiego entuzjazmu dla nowej zabawki tylko czekały na zorganizowane wykorzystanie, nie otrzymywali oni nawet schematu połączeń wewnętrznych ani podstawowych informacji o systemie operacyjnym. Postępowanie takie nie było przypadkowe: MERITUM powstawało poza wszelkimi planami centralnymi i poza wszelkimi dodatkowymi środkami na prace badawcze i rozwojowo-wdrożeniowe. Zespół autorski w składzie dwóch dyrektorów i jednego głównego konstruktora mógł bardzo pomóc swemu dziecku na terenie fabryki, lecz przecież nie mógł zastąpić laboratorium przemysłowego z prawdziwego zdarzenia. W konsekwencji w sensie programowym MERITUM jest po prostu dokładną (co do bitu) kopią TRS-80, co ułatwia przenoszenie programów, ale też pociąga za sobą paradoksy: ten polski produkt przeznaczony dla polskiej oświaty nawet gotowość do pracy zgłasza po angielsku (w wersji MERITUM 1.1)! Tak dalece posunięta nieprzypadkowa zgodność z wzorcem musiała budzić u producenta pewien niepokój, czy aby nie dojdzie do przebudzenia się niedźwiedzia i czy TRS nie zacznie dochodzić swych praw autorskich. Mimo więc zakupienia zawartości ROM od firmy ITM woleli oni nie chwalić się, co w tym komputerze siedzi.

Ta polityka srodze się zemściła: do dziś nie powstał na większą skalę rynek intelektualny na programy i udoskonalenia MERITUM. Historia pokazała, że nawet IBM do sukcesu potrzebuje wsparcia otoczki konkurentów-pomocników (rozwijających za własne pieniądze i na własne ryzyko produkt, którego dokumentacja techniczna jest w pełni dostępna).

Popełniane w propagowaniu wyrobu błędy miały i bardziej prozaiczne przyczyny: w MERA-ELZAB, jak i w całym polskim przemyśle komputerowym, nigdy wcześniej nie produkowano wyrobów masowych. Pion handlowy przedsiębiorstwa i jego służby techniczne nastawione były na obsługę najwyżej kilkudziesięciu klientów, z których każdy kupuje za miliony, każdego można poznać osobiście i z których każdy zna się na informatyce i wie czego chce. Nie trzeba więc ich do niczego agitować, sami złożą zamówienie i będą pokornie czekać w kolejce.

* * *

Do kroniki: pierwsze informacje o MERITUM ukazały się latem 1983 r., jego pierwsza publiczna prezentacja – jesienią tegoż roku, montaż seryjny rozpoczęto latem 1984 r., a do lata 1985 r. powstało tysiąc kilkaset egzemplarzy modelu MERITUM I.1. Od tego czasu rozpoczęto produkcję modelu MERITUM I.2., wyposażonego – przy nie zmienionej cenie! – w pełne 64 KB RAM oraz polskie znaki na klawiaturze. Jesienią 1985 r. rozpoczęto produkcję MERITUM II wyposażonego w stację dysków elastycznych 5,25 cala z dwoma napędami produkcji NRD i system operacyjny CP/M 2.2. Producentowi udało się pozyskać cennego sojusznika, dyr. Edwarda Solarskiego z Centrum Obliczeniowego Politechniki Śląskiej, w którym opracowano większość dostępnego profesjonalnego oprogramowania dla MERITUM: bardzo dobre programy typu kartoteka i edytor tekstów, kompilatory, interpreter LOGO itp. Ośrodek ten również masowo tłumaczy na MERITUM programy rozrywkowe ze Spectrum.

* * *

Drugim propagatorem i sojusznikiem MERITUM jest dyrektor naczelny zrzeszenia MERA Henryk Piłko, były dyrektor Elzabu, który nie ustaje w namawianiu oświaty na zakup MERITUM. Ufundował on kilkunastu szkołom pracownie mikrokomputerowe pełne MERITUM, zorga-

nizował w Warszawie centrum kompletacji mikrokomputerów, nastawione na obsługę i wspieranie niefachowych klientów, starał się zamówić w różnych miejscach oprogramowanie i podstawowe materiały pomocnicze: instrukcje, poradniki itp.

Niestety, jak dotąd wysiłki te, czynione z wielką energią, napotykają równie wielki brak wiary otoczenia w powodzenie całej sprawy. Zainteresowanie najbardziej twórczych fanatyków mikrokomputerów przesunęło się już chyba nieodwracalnie w stronę kolejnych generacji sprzętu: IBM PC, Amstradów, nowszych modeli Commodore i Atari. Spectrum, Commodore 64 i Atari 800 XL mają wciąż jeszcze swych zaprzysięgłych zwolenników, ale gdzie szukać zapaleńców, gotowych włożyć całe swe siły w opisywanie i wzbogacanie MERITUM? Bez nich, własnymi siłami, przemysł może udoskonalić i wyprodukować sprzęt. Stworzenie dla niego infrastruktury intelektualnej okazuje się trudniejsze.

* * *

Powstaje więc sytuacja paradoksalna: oto po dwóch latach stopniowego rozwoju i doskonalenia krajowego wyrobu mamy dziś całkiem przyzwoity mikrokomputer domowy po zupełnie racjonalnej cenie. Zbudowana z niego sieć lokalna opracowana w Politechnice Śląskiej (szczegółowy opis przedstawimy za miesiąc) z procesorami graficznymi i wspólnie przez wszystkich użytkowników wykorzystywaną drukarką oraz CP/M 2.2, pamięcią masową na dyskach elastycznych – jest obecnie najtańszą (ok. 4 mln za 10 stanowisk) tego rodzaju, dostępną na naszym rynku propozycją, z przyzwoitym oprogramowaniem i zadowalającą niezawodnością. Jest... a przynajmniej mogłaby być.

Mogłaby być, gdyby... nie sukcesy eksportowe Zabrza i kilka innych niemożności. W połowie 1985 r. dyrekcja zakładów Elzab stanęła bowiem przed dylematem: czy przyjąć niezwykle atrakcyjne i dochodowe zamówienia radzieckie na podstawowy wyrób firmy (terminale) w skali przekraczającej dotychczasowe możliwości produkcyjne, czy nadal rozpraszać siły na mikrokomputery? Zwyciężył podstawowy interes zakładu i dla mikrokomputerów zaczęło brakować miejsca w fabryce i czasu u dyrektorów. W ramach zrzeszenia uzgodniono, że produkcja MERITUM będzie stopniowo przenoszona do jednej z filii ELWRO. Dla ELWRO MERITUM jest jednak po trochu podrzutkiem – dyrektorzy tego zakładu mają swoje własne ukochane dzieci: ELWRO 800 z rodzinką. ELWRO przez pewien czas udawało, że deklaruje, iż rozważa MERITUM jako jedną z wchodzących w grę propozycji przyszłego mikrokomputera szkolnego produkowanego w ramach zamówienia rządowego, ale z czasem deklaracje te stawały się coraz bardziej mgliste. Stworzony przez własnych konstruktorów zakładu SOLUM (który podobno ma już grafikę) oraz skonstruowane w Instytucie Automatyki Politechniki Poznańskiej ELWRO 800 Jr bliższe są wrocławskim sercom.

Biednemu wiatr w oczy wieje: przez kilka miesięcy produkcja MERITUM II była wstrzymana ze względu na brak napędów dyskowych, nie udało się dotąd uruchomić przemysłowego producenta procesorów graficznych, w końcu Politechnika Śląska będzie je robić we własnym zakładzie doświadczalnym. Tymczasem mijają miesiące, a wymagania społeczne wobec mikrokomputerów rosną. Zabawki te przestają być aż tak niespotykane i towar, produkowany w skali mniejszej niż 10 tysięcy egzemplarzy rocznie, nie ma szans, ginie w tłumie. W tym roku powstanie 4 tys. sztuk MERITUM.

Co dalej z MERITUM? Jak dotąd klientów nie brakuje, portfele zamówień są wypełnione. Dla wielu przedsiębiorstw jest to wciąż najtańszy i najatrakcyjniejszy sposób zakupu gotowej płyty mikrokomputera Z80 z obudową i klawiaturą, płyty, z której we własnym zakresie tworzy się potem własny specjalizowany system. Wiele innych instytucji marzy o taniej i prostej sieci lokalnej do zastosowań, typu małej biurokracji i lokalnej biurowej wymiany informacji, zastosowań przynoszących nam wszystkim ulgę i poprawę komfortu obsługi, tempa załatwiania spraw.

AGAPOL Pawła Podsiadły podbija rynek krajowy i RWPG urządzeniami specjalizowanymi opartymi na MERITUM: stworzony przezeń komputerowy kartometr jest dziś podstawowym narzędziem każdego kartografa i geodety. Wiele przedsiębiorstw chwali sobie systemy kierowania urządzeniami produkcyjnymi sterowane przez MERITUM. Należy do nich m.in. Huta Warszawa, w której ten najprostszy sprzęt z powodzeniem steruje wytopem stali.

Wszystko idzie jak zwykle, ani gorzej, ani lepiej – ale czas MERITUM powoli przemija. Warszawska czy wrocławska młodzież jeszcze przez wiele lat kontakt z informatyką zaczynać będzie od tego urządzenia, marząc jednak o nowszych, bardziej atrakcyjnych maszynach.

WŁADYSŁAW MAJEWSKI

Dyrektor Centrum Obliczeniowego Politechniki Śląskiej, mgr Edward Solarski, prezentuje sieć lokalną MERITUM skonstruowaną przez jego ośrodek podczas I Światowej Wystawy Młodych Wynalazców w Płowdiwie, listopad 1985 r.



MIKROGALA

I międzynarodowa wystawa komputerów osobistych, domowych i biurowych, zorganizowana w dniach 8-10 kwietnia w hotelu "Victoria" w Warszawie, zgromadziła wielu liczących się na rynku światowym producentów sprzętu komputerowego i urządzeń peryferyjnych. Przeżywające okres stagnacji firmy zachodnie wyraźne wyjście z sytuacji upatrują w eksporcie na rynki socjalistyczne. Przemysł krajowy zainteresowania imprezą nie wykazał: oprócz Mery-Błonie żadnego wystawcy państwowego w "Victorii" nie było.

Dla nas wystawa była okazją do rozeznania intencji i ocen. Zdając sobie sprawę z tego, że nadmierne absorbowanie handlowców teoretycznymi dociekaniami nie jest na żadnej imprezie o charakterze handlowym mile widziane – zdecydowaliśmy się na przeprowadzenie wśród najważniejszych i liczących się na światowym rynku producentów krótkiej ankiety. Oprócz australijskiej filii koncernu IBM, której odpowiedź na cztery pytania zająć miała – bagatela – tydzień, wszyscy pozostali wypełnili naszą ankietę praktycznie od ręki.

1. Jakie nadzieje firma wiązała z udziałem w tej wystawie i na ile nadzieje te spełniły się?
2. Co ze swych najnowszych osiągnięć prezentowaliście?
3. Jaką ofertę w zakresie ewentualnej kooperacji z polskim przemysłem elektronicznym jesteście w stanie przedstawić?
4. Jak oceniacie warszawską wystawę?

ICL

Z racji wieku (od 20 lat obecni na polskim rynku) pierwszeństwo przysługuje brytyjskiemu koncernowi INTERNATIONAL COMPUTERS LIMITED:

Ad 1.

Podejmując decyzję o udziale w warszawskiej wystawie kierowaliśmy się głównie myślą o promocji naszego sprzętu typu PC i systemów biurowych. Dotychczas firma znana była w Polsce jedynie jako producent dużych systemów.

Ad 2.

- Supermikrokomputer "ICL CLAN", oparty o 32-bitowe mikroprocesory i pracujący w systemie operacyjnym UNIPLUS+ będącym praktycznie odpowiednikiem UNIX-a.
- Komputer osobisty OPD (One Per Desk), stanowiący połączenie telefonu i PC.
- System Zasobów Rozproszonych DRS20, stanowiący połączenie techniki mikroprocesorowej, systemów wieloprocesorowych i komputerowych sieci lokalnych LAN.

Ad 3.

Nasza firma złożyła ofertę na koprodukcję komputerów osobistych. (Niestety rozmowy na ten temat otoczone są tajemnicą handlową – Red.).

Ad 4.

Oceniając imprezę podkreślić należy bardzo duże zainteresowanie eksponowanym sprzętem. W minionych latach firma ICL "komputeryzowała" Bank Handlowy, Polską Agencję Prasową, Główny Urząd Statystyczny, liczne zakłady przemysłowe, a jej obroty z Polską osiągnęły w 1985 r. wartość 2 mln funtów. Prezentowany w Warszawie supermikrokomputer CLAN zasługuje na odrębne omówienie. 32-bitowa maszyna oferowana jest do sprzedaży do Polski w sytuacji, gdy licencjonowany jest eksport do nas komputerów 16-bitowych. Jedna z wielu możliwości wykorzystania OPD – to np. przekazanie przez komputer drogą telefoniczną zapisanego w jego pamięci tekstu rozmówcy po drugiej stronie linii – oczywiście dzięki wbudowanemu syntezerowi mowy. O określonej porze komputer sam wybierze numer, a po uzyskaniu połączenia przekaże wiadomość.



Obliczy Pobawi Doniesie



Trzy cale w ofensywie – timexopodobny 2086

W VICTORII

HEWLETT PACKARD

Znana na polskim rynku od 15 lat amerykańska firma HEWLETT-PACKARD kojarzy się na ogół przeciętnemu zjadaczowi chleba z kalkulatorami. W "Victorii" kalkulatorów z charakterystycznym znakiem HP nie było. Przedstawiciele ZOTPAN-u, który od 10 lat prowadzi autoryzowany serwis firmy, przedstawiali sprzęt nieporównywalnie bardziej zaawansowany. Na ankietę odpowiedzieli:

Ad 1.

– Nasza obecność tutaj wiąże się z nadzieją na wzrost sprzedaży produktów HEWLETT-PACKARDA na polskim rynku. O tym, czy nadzieje te spełniły się, można będzie powiedzieć dopiero po pewnym czasie. Jednakże, tak jak spodziewaliśmy się, zainteresowanie naszymi produktami było bardzo duże.

Ad 2.

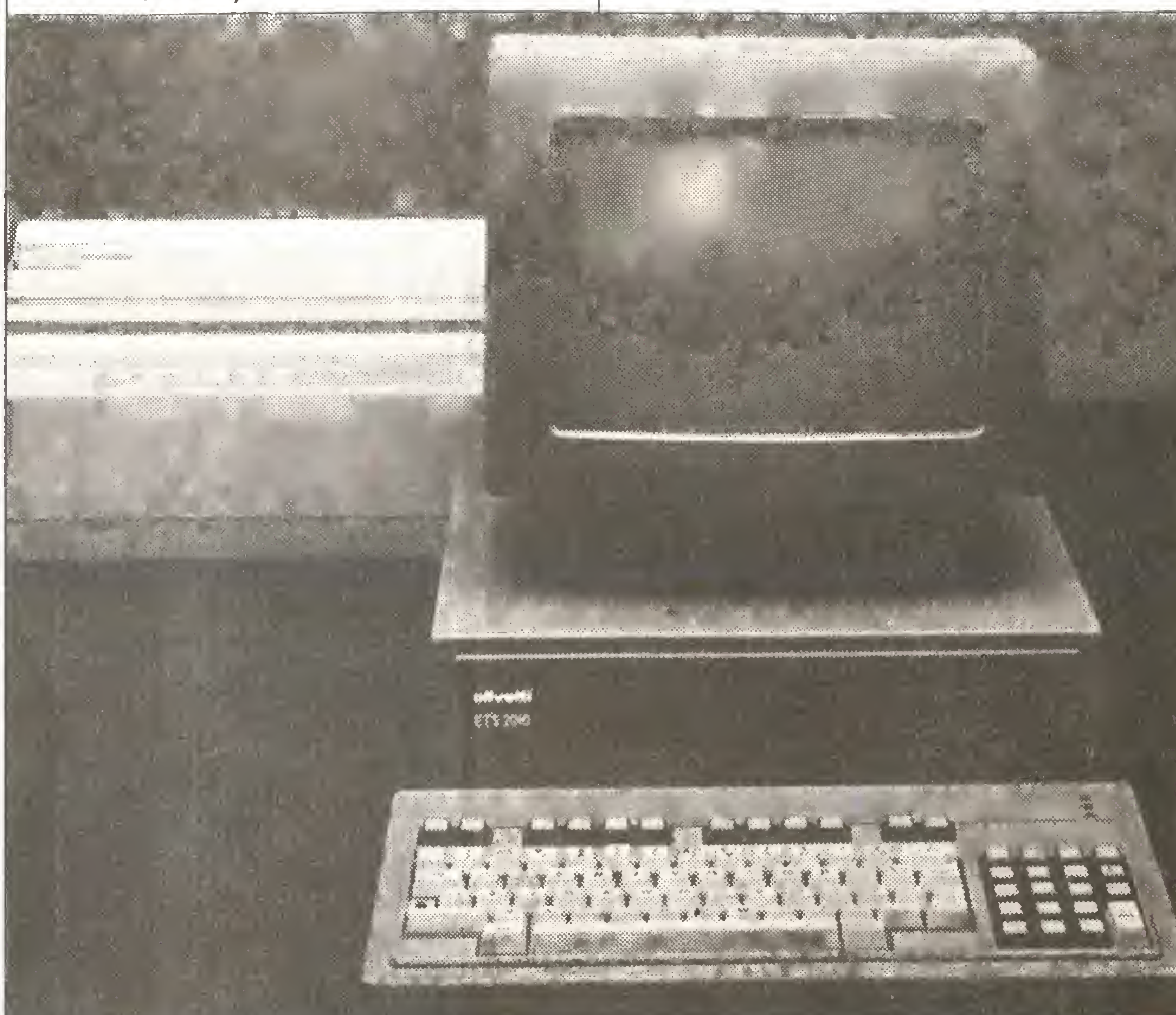
- Drukarka laserowa HP 2686 A/B
- Plottery formatu A1 i A3
- Komputery osobiste HP 150II oraz HP 9816S
- 20-megabajtowa pamięć dyskowa (dysk typu Winchester) HP 9133D

Ad 3.

– Plottery HEWLETT-PACKARDA mogą współpracować z komputerami polskiej produkcji i istnieje w związku z tym możliwość oferowania ich w zestawach z polskimi komputerami na rynkach trzecich.

Ten amerykański koncern jest jednym z potentatów w dziedzinie automatycznych systemów sterowania produkcją. Fabryki HP rozmieszczona są nie tylko w USA, lecz również w Wielkiej Brytanii, RFN, Francji a nawet Japonii. Na rynku polskim oferuje sprzęt w następujących dziedzinach: duże systemy komputerowe, komputery osobiste do zastosowań profesjonalnych, programowane kalkulatory kieszonkowe, elektroniczna aparatura pomiarowa i analityczna oraz sprzęt medyczny. Eksport do Polski przekroczył w ub. r. wartość 1,5 mln dol.

olivetti



Marzenie redakcji, cały numer „KOMPUTERA" na dyskietce 5,25 cala



Naszą ankietą nie objęliśmy licznie reprezentowanych na wystawie firm polonijnych i rzemieślniczych, m.in. Studia Komputerowego Kajkowscy i COMPUTEX-u, oferujących interesujące polskojęzyczne programy redagujące dla IBM PC i wiele innych programów. Polbrit przedstawiał znany już programowo zgodny z Timexem 2068 Unipolbrit wraz ze sterowaną własnym mikroprocesorem stacją dysków 3 lub 2,25 cala, tworzącą z Unipolbritem system chodzący pod CP/M.

Apina, jedyny dziś w Polsce dostawca drukarek za złotówki w rozsądnych terminach proponuje też własną wersję stacji dysków 5,25 cala do Spectrum oraz samo Spectrum Plus. Na wyroby tych firm wciąż utrzymuje się wysoki popyt, mimo cen obliczonych raczej na instytucje niż prywatnych klientów.

Znak firmowy OLIVETTI znany jest w naszym kraju głównie z maszyn do pisania. Na rynku polskim Włosi osadzeni są dosyć mocno. Stąd też na pierwsze pytanie naszej ankiety odpowiedzieli:

Ad 1.

– Mamy nadzieję na pozyskanie nowych klientów. Ponadto przedstawiamy informację i prezentujemy nowy sprzęt.

Ad 2.

- Komputery osobiste M 24 PC i M 21 PC
- System przetwarzania tekstów w zastosowaniach biurowych ETS 2010
- Elektroniczne maszyny do pisania ET 115 i 111 z polską klawiaturą

Ad 3.

– Wystawy tego typu powinny być organizowane corocznie.

Pominięcie odpowiedzi na 3 pytanie wydaje się świadczyć o tym, że OLIVETTI zainteresowana jest polskim rynkiem wyłącznie jako dostawca sprzętu. A szkoda, gdyż "Polska Olivetti" w "Polskim Fiacie" polskiego dziennikarza – to dla nas, dziennikarzy, perspektywa zachęcająca. Na razie musimy się zadowolić kontemplacją wspomnianego systemu przetwarzania tekstów. Szerzej o podobnym urządzeniu (Amstrad 8256) pisaliśmy w poprzednim numerze "Komputera". Ograniczymy się więc jedynie do informacji, że ETS 2010 ma również 256 KB pamięci, ale jej serce stanowi nie Z 80, a 16-bitowy INTEL 80186. Nośnikami informacji są 2 floppy dyski 5 i 1/4 cala.

VICTORIA W

Japończycy reprezentowani byli w "Victorii" przez swe zachodnioniemieckie przedstawicielstwa handlowe. STAR wystawiał szereg drukarek, w produkcji których się specjalizuje, a EPSON – oprócz drukarek, również komputery osobiste i przenośne (portable) z ekranami na ciekłych kryształach.

O ile jednak STAR znany jest na rynku polskim nie tylko ze słyszenia, lecz również z drukarek sprzedawanych indywidualnym użytkownikom dysponującym odpowiednio zasobnym kontem A przez bońską firmę wysyłkową ABC-Data GmbH (w reklamówce ważna adnotacja – "mówimy po polsku"), o tyle EPSON dopiero rozeznaje nasz rynek. A nuż też się zdecydują!

Na pytania naszej ankiety przedstawiciele obu firm odpowiedzieli:

Ad 1.

– Celem naszej ekspozycji jest bezpośredni kontakt z użytkownikami drukarek STAR w Polsce, stymulowanie ich sprzedaży i stosowne rozreklamowanie naszego serwisu po- i gwarancyjnego.

Ad 2.

– Eksponowane przez nas drukarki NL-10 oraz NB-15 (24-igłowa) przedstawiają sobą najnowsze osiągnięcia techniki w dziedzinie drukarek mozaikowych.

Ad 3.

– Bardzo dobrze układa się nasza współpraca kooperacyjna z Metronexem i zakładami Mera-Błonie.

Ad 4.

– Taka wystawa musi być organizowana corocznie!

– Oferta STARA jest dla nas szczególnie interesująca. Wycofywany już model Star GEMINI 10X od dwóch lat powszechnie uchodził za najlepszą propozycję dla domowego użytkownika: łączy on relatywnie niską cenę (poniżej 200 dol. w USA, ok. 600 marek w RFN) z wysoką niezawodnością i dobrą jakością druku przy standardowej (A4) szerokości papieru. Zadowala się zwykłym papierem maszynowym oraz zwykłą taśmą maszynową o szerokości 13 mm (do nabycia w każdym sklepie papierniczym). Pod tym wzglę-



star
Gedruckt mit IBM Steckmodul

STAR NL-10

Doppelt hoch dnd breit in NLQ und EDV-Druck

STAR NL-10 STAR NL-10

Vierfach hoch und breit in NLQ und EDV Druck

STAR NL-10STAR NL-10

Auto Centering,d.h. eine Zeile kann rechts-, linksbundig oder zentriert gedruckt werden.

5 KByte Speicher serienmäßig

- Einstellen verschiedener Funktionen am Bedienungsfeld moglich, wie:
  - Umschaltung NLQ – EDV Druck
  - linker/rechter Rand
  - Zeichenbreite
  - Fettdruck
  - Microlinefeed vorwarts wie rückwarts

- halbautomatischer Einzelblatteinzug
- automatischer Einzelblattzuführer (Option)
- durch Schubtraktor Papiertransport vorwärts und rückwarts
- benutzerdefinierbare Zeichen in NLQ-, EDV- und Proportionaldruck
- Papierabrißkante
- Proportionalschrift
- Farbbandkassette
- 9 Nadel Druckkopf
- 120 CPS (30 CPS in NLQ)
- HEX Dump

VERSCHIEDENE INTERFACE-STECKMODULE (IBM, C 64, PARALLEL usw.)

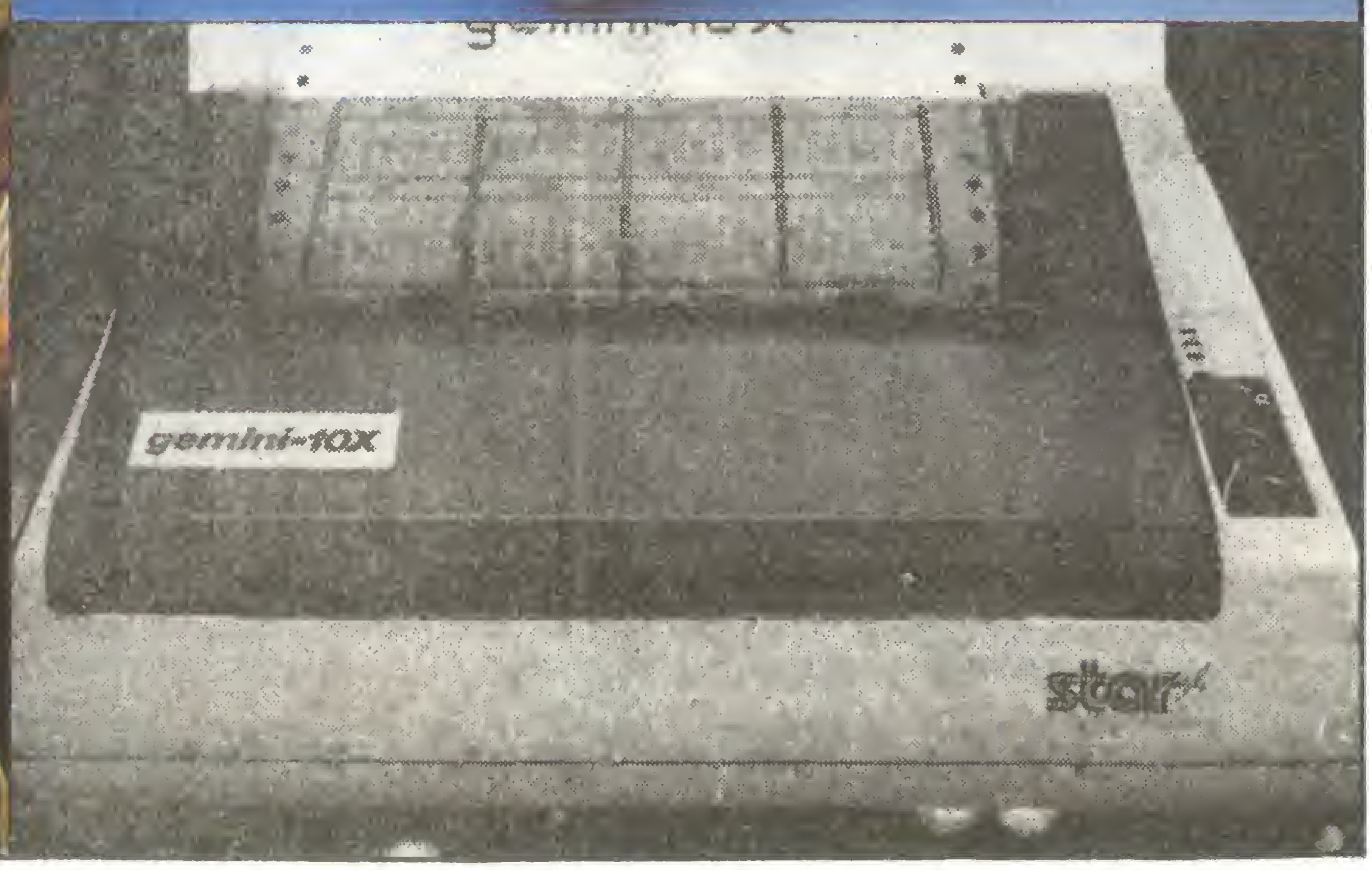
SUPERSCRIPT oder SUBSCRIPT in einer Zeile

Internationale Zeichensatze:

Zeichensatz Nr.: 0	#$@[\]^`{¦}~	U.S.A.	
Zeichensatz Nr.: 1	#$à°ç§^`éùè¨	FRANKREICH	
Zeichensatz Nr.: 2	#$§ÄÖÜ^`äöüß	DEUTSCHLAND	
Zeichensatz Nr.: 3	£$@[\]^`{¦}~	ENGLAND	
Zeichensatz Nr.: 4	#$@ÆØÅ^`æøå~	DANEMARK 1	
Zeichensatz Nr.: 5	#¤ÉÄÖÅÜéäöåü	SCHWEDEN	
Zeichensatz Nr.: 6	#$@°\é^ùàòèì	ITALIEN	
Zeichensatz Nr.: 7	₧$@¡Ñ¿^`ñ}~	SPANIEN	
Zeichensatz Nr.: 8	#$@[¥]^`{¦}~	JAPAN	nur über Software ansteuerbar
Zeichensatz Nr.: 9	#¤ÉÆØÅÜéæøåü	NORWEGEN	nur über Software ansteuerbar







MIKROGALI

dem Star GEMINI 10X bije na głowę nawet krajową drukarkę D-100, która wymaga praktycznie niedostępnych w kraju kaset z taśmą bez końca ("ribbon"), kosztujących za granicą kilka dolarów sztuka. Przy intensywnej eksploatacji taśma taka wymaga wymiany co kilka tygodni. Jest to dla przeciętnego polskiego użytkownika zaleta rozstrzygająca, tak więc informację o zakończeniu produkcji tego modelu większość zwiedzających przyjmowała z wielkim żalem. Dodać trzeba, że model ten cechowała także prostota i łatwość dostosowania do współpracy z większością popularnych mikrokomputerów.

Następcą GEMINI 10X jest STAR 10 NL, którego test redakcyjny – dzięki uprzejmości firmy – przedstawimy w jednym z najbliższych numerów. Niestety drukarka ta wymaga specjalnych kaset z taśmą.

Ad 1.

– Ponieważ dopiero chcemy robić interesy w Polsce, decyzję o udziale w tej wystawie podjęto w nadziei na prezentację naszych możliwości z jednej, a na rozeznanie o rynku, o którym słyszeliśmy, że jest bardzo chłonny – z drugiej. Czy i na ile nadzieje te były uzasadnione – pokaże czas.

Ad 2.

- 16-bitowy komputer osobisty EPSON PC
- Drukarka mozaikowa LQ-1000

Ad 3.

– Polityka handlowa prowadzona przez naszą firmę nie przewiduje możliwości kooperacji.

Ad 4.

– Uważamy, że wystawa była dobrze zorganizowana. Zdecydowana większość zwiedzających wiedziała czego chce. Treściwe pytania świadczyły o tym, że niewielu było tu przypadkowych przechodniów.

EPSON

24-igłowa mozaikowa drukarka LQ-1000 pracuje z szybkością 180 (grafika) lub 60 (litery) znaków na sekundę i reklamowana jest jako ostatnie słowo Japończyków.

LQ 1000 w trybie graficznym oznacza 14 x 7 punktów na mm



16 bitowa maszyna bez kompromisów



Półtora kg elektroniki



JX 80 – tęcza spod igieł



MIKROGALA

Mera-Błonie

Jedynym państwowym przedsiębiorstwem broniącym polskich barw na warszawskiej wystawie była MERA-BŁONIE. Na razie na dostawę drukarek z Błonia czekać trzeba i półtora roku. Ponoć jeszcze trochę i również na rynek krajowy D-100 popłyną szerszym strumieniem.

Przedstawiciel Błonia stwierdził, że:

Ad 1.

– Celem naszego udziału było zapoznanie potencjalnych klientów z ofertą zakładu. Pragniemy również nawiązać kontakty kooperacyjne z firmami z państw kapitalistycznych. Zakład oferuje im swoją technologię do produkcji podzespołów stosowanych rutynowo w sprzęcie komputerowym.

Ad 2.

● Rodzina drukarek D-100, w tym D-100E (kompatybilna z FX-80 Epson).

Ad. 3

– Drukarka D-100 E/PC do polskiego IBM PC, którego produkcję uruchamia spółka "Mikrokomputery". MERA-BŁONIE jest udziałowcem tego przedsięwzięcia.

Ad. 4

– Nie podejmujemy się oceniać – brak kryteriów.



Drukarka D-100 mozaikowa



Stoisko firmy wysyłkowej POLANGLIA Ltd

Najnowszą ofertę handlową firmy wysyłkowej POLANGLIA zamieszczamy na stronie 48

Eufemistycznie sformułowana przez przedstawiciela MERY ocena imprezy wymaga – na zakończenie – pewnego rozwinięcia. Jej organizację należy ocenić bardzo krytycznie. Ciasne pomieszczenia wystawowe nie tylko ograniczyły możliwości ekspozycji, lecz wręcz uniemożliwiały poruszanie się zwiedzającym. Tłum, jaki przez trzy kolejne dni walił do tego najlepszego – było nie było – hotelu w Warszawie, był segregowany i ustawiany w długiej kolejce, a kontrola zaproszeń oraz dowodów osobistych (sic!) dokładnie odsiewała wszystkich nieupoważnionych i liczących poniżej 18 lat. Pierwszego dnia zwiedzającym i zwiedzanym we znaki dała się temperatura i duchota ponoć klimatyzowanych pomieszczeń. Na szczęście 9 i 10 kwietnia aura już nie dopisała. Miejmy nadzieję, że za rok organizatorzy (AGPOL) wyciągną wnioski i na zapowiadanej już kolejnej wystawie komputerów osobistych, domowych i biurowych będzie lepiej i bardziej reprezentatywnie (broń Boże reprezentacyjnie).

Ankietował, skwapliwie notował
i do druku przygotował:
MAREK CAR



PROGRAMY, NA KTÓRE CZEKAMY

Przedstawiamy nowe i najnowsze programy, przeznaczone dla komputera ZX Spectrum, które dotarły już do naszego kraju lub takie, które (mam nadzleję) wkrótce dotrą.

Długo oczekiwane "COMMANDO", "SABOTEUR" są już w Polsce. Mamy też kontynuacje popularnych wcześniej gier: "BEACH HEAD 2", "JET SET WILLY 2" i "BACK TO SKOOL" (druga część "SKOOL DAZE"). Czy zawiodły one nasze oczekiwania? Czekamy na uwagi i opinie naszych Czytelników o tych programach (o innych oczywiście także). Zagraniczne czasopisma często publikują listy najbardziej popularnych programów i jeżeli otrzymamy Wasze oceny, również postaramy się zestawić i opublikować podobną listę.

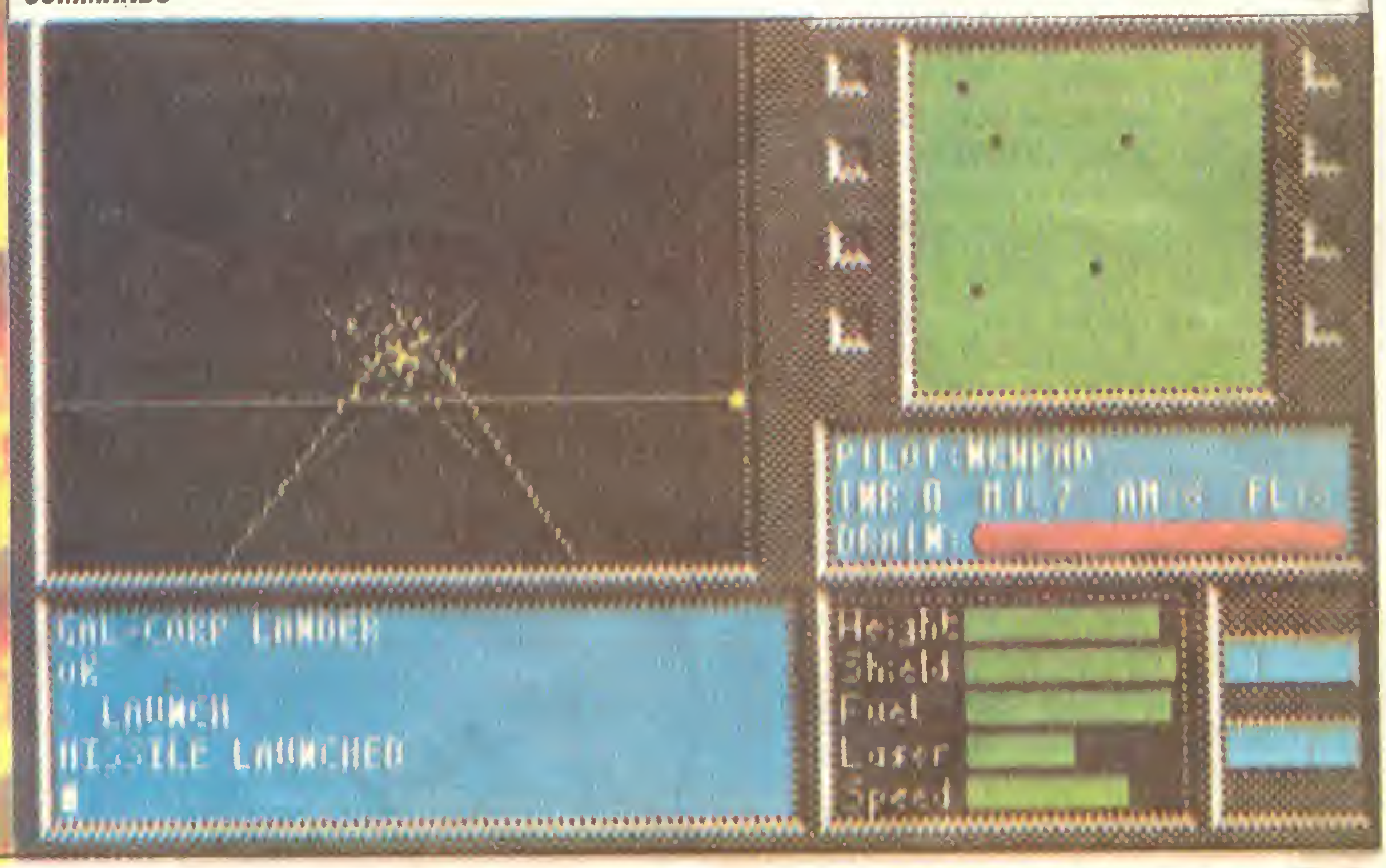
Do nowości zaliczamy "THREE WEEKS IN PARADISE", "TAU CETI", "INTERNATIONAL KARATE"

▶ 22



COMMANDO

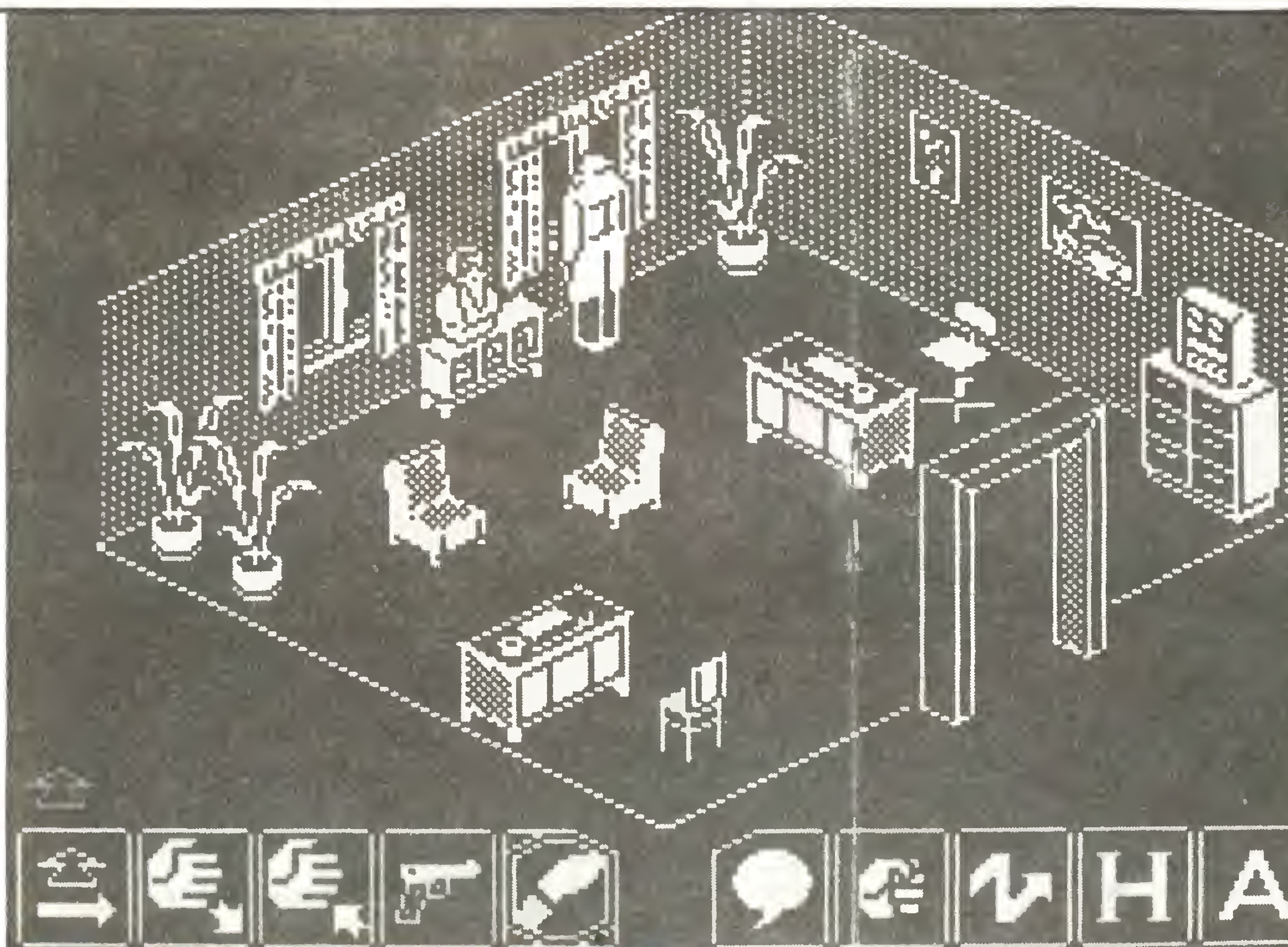
TAU CETI



I "YIE AR KUNG FU". Te programy już są w kraju, choć nie były poprzedzone dużą reklamą. Ten brak reklamy powoduje, że niektóre wartościowe i ciekawe programy pojawiają się z dużym opóźnieniem lub wcale. Przykładem są "DYNAMITE DAN" (jest już reklamowana druga wersja) i bardzo już stary "RAPSCALION".

Kontynuacją stylu znanej nam już gry "FAIRLIGHT" są "MOVIE" firmy IMAGINE-OCEAN i "SWEEVO'S WORLD" firmy Gargoyle Games. Znana z dobrych i często nowatorskich pomysłów ULTIMATE opublikowała trzy nowe tytuły "GUNFRIGHT", "CYBERUN" i "PENTAGRAM".

Pamiętamy sympatycznego Charlie'ego z "NODES OF YESOD". Pojawił się znów w "ARC OF YESOD" firmy THOR-ODIN. Ta sama firma przedstawia ponadto nową grę o dziwnym tytule "I.C.U.P.S.". Odin (tym razem samodzielnie) opublikowała "ROBIN OF THE WOOD", którym



SWEEVO'S WORLD





ENIGMA

MOVIE

cieszą się nieliczni jeszcze posiadacze w Polsce.

W drugim numerze opisywaliśmy interesującą grę "SHADOWFIRE". Jej wydawca, firma Beyond przedstawiła już drugi z tego cyklu program – "ENIGMA FORCE". Nie posiadamy również starszego produktu tej firmy: "SPY US SPY".

Bardzo reklamowana "ELITE" opublikowana przez FIREBIRD (z dołączonym słynnym już LENSLOK'iem utrudniającym życie "włamywaczom") również czeka na sprowadzenie. "TOMAHAWK", też z LENSLOK'iem, jest już w kraju i rodzimi włamywacze łatwo dali sobie radę z zabezpieczeniem (interface MUEL 85 okazał się bardzo skuteczny). Nie taki więc LENSLOK straszny...

Mikro-Gen to gwarancja dobrej zabawy i jeszcze lepszej grafiki. Oczekiwany "SIR FRED", który trochę przypomina "JET SET WILLY", nie powinien zawieść naszych oczekiwań. Podobne walory cechują produkty Gremlin Graphics, której "MONTY ON THE RUN" jeszcze do nas nie dotarł.

Czeka na sprowadzenie edytor tekstu "TASWORD 3", przeznaczony głównie dla użytkowników interface 1 i microdrive'u. Parametry tego programu znacznie przewyższają znaną nam już wersję i myślę, że możliwe jest przystosowanie go do współpracy z magnetofonem.

Grafików powinien zainteresować program firmy OCP "ART STUDIO", brydżyści z pewnością czekają na "BRIDGE PLAYER 3", zaś "LASER BASIC" może okazać się pomocny przy tworzeniu własnych gier zręcznościowych. "SUPERCODE 3" powinien okazać się pożyteczny dla wszystkich.

Należy jeszcze wspomnieć o innych nowych grach: Imagine – "MIKIE", znanej firmy Quicksilva "MAX HEADROOM, a także "ZOIDS" – wynik pracy programistów z Martech oraz "SPELLBOUND" z Mastertronic.

Gry przygodowe mimo trudności językowych (czekamy na grę przygodową po polsku) mają liczną grupę sympatyków, którzy teraz z pewnością siedzą z wypiekami na twarzy nad "LORD OF THE RINGS" firmy Melbourne House. Program oparty jest na słynnej trylogii Tolkiena "Władca pierścieni", kontynuując znaną wszystkim grę "HOBBIT". Powinni jednak pamiętać o "TERROMOLINOS" tej samej firmy, a także o "NEVER ENDING STORY" opublikowanej przez OCEAN.

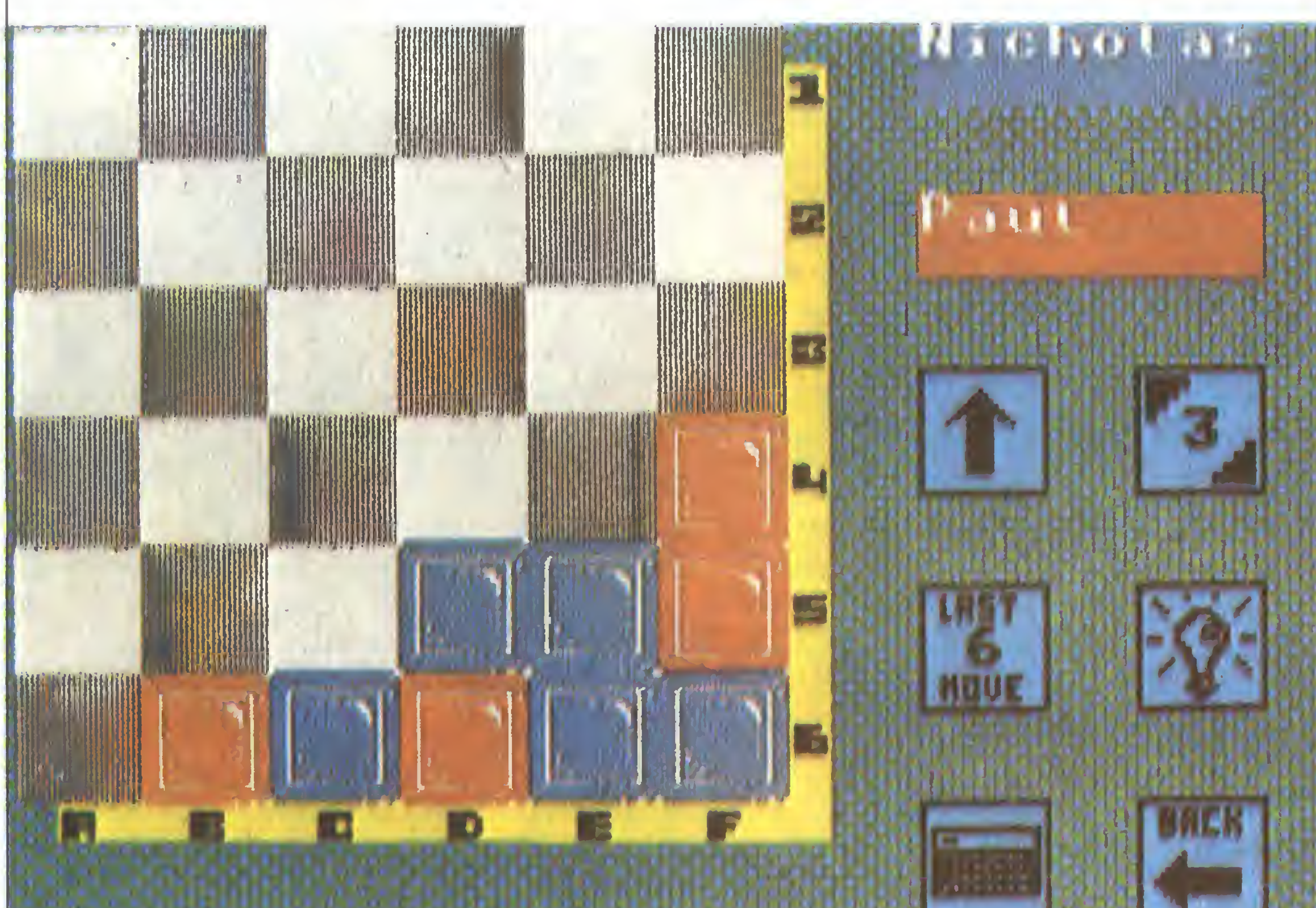
Firma LEVEL 9 znana jest z bardzo dobrych programów przygodowych, które z reguły zajmują pierwsze miejsca na "listach przebojów". "EMERALD ISLAND", "A SECRET DIARY OF ADRIAN MOLE" i najnowszy "WORM IN PARADISE" wciąż czekają na sprowadzenie.

"SWORDS AND SORCERY" firmy PSS przedstawiana jest jako najlepszy komputerowy odpowiednik gier role playing, reklamowanych przez bratnią redakcję "Fantastyki".

"THINK!", czyli "myśl!" czeka na wielbicieli planszowych gier strategiczno-logicznych. Proste zasady i skomplikowana strategia zapewnią wiele godzin rozkoszy łamania głowy.

GRZEGORZ CZAPKIEWICZ

THINK!



MARS

Komputer: ZX Spectrum 48K
Wydawca: Gargoyle Games, 1985
Autor: Greg Follis, Roy Carter

Akcja "Marsportu", pierwszej gry z zapowiadanej trylogii, rozgrywa się w okresie, w którym Ziemianie mają poważne kłopoty z Septami – agresywną, niesympatyczną rasą kosmitów. Ziemia i Księżyc chronione są przed nimi przez gigantyczną sferę pola siłowego. Kłopot polega na tym, że Septowie odkryli sposób na jej przedziurawienie. Oryginalne plany, zawierające również – pozostawione przez nieżyjącego już konstruktora sfery – szkice, dotyczące jej przekonstruowania i wzmocnienia spoczywają w pamięci centralnego komputera Marsportu. I tylko tam. Zaś Mars jest w rękach (czy raczej odnóżach) Septów.

Zasiadając przy klawiaturze ZX Spectrum stajesz się, Graczu, innym człowiekiem. Nazywasz się teraz John Marsh i jesteś komandorem sił ziemskich. Twoje zadanie to spenetrowanie podziemnego labiryntu Marsportu, odnalezienie centralnego komputera, odzyskanie planów i oczywiście bezpieczna ucieczka. Przeszkadzać Ci w tym będą nie tylko Septowie, ale również, a może nawet przede wszystkim, centralny komputer i zaprojektowany przez niego w odruchu samoobrony system ochronny. Twoim pierwszym celem w Marsporcie winno być znalezienie broni i doprowadzenie jej do stanu używalności. Bezbronny nie masz żadnych szans.

Korytarze Marsportu patrolowane są przez Septów-wojowników, dwukrotnie od Ciebie niższych i zdających się nie posiadać rażącej na odległość broni. To jednak wcale nie znaczy, że niegroźnych. W niektórych przejściach, łączących sektory tego samego poziomu miasta, natkniesz się na pozornie nieruchomych Septów-wodzów. Dobra rada – raczej nie podchodź do nich zbyt blisko. Pozostają jeszcze patrole sterowanych przez centralny komputer robotów. Wydawałoby się, iż powinny Cię one witać radosnym mruganiem wszystkich światełek. Niestety, niektóre z nich biorą Cię za Septa i starają się zrobić z Tobą to, co komputer każe im robić z tymi kosmicznymi najeźdźcami. Tak więc sam widzisz – czeka Cię niełatwe zadanie.

Jednak droga do sukcesu w "Marsporcie" nie polega na strzelaniu na prawo i lewo z blastera. Przede wszy-

POKE n,∞

Wszyscy lubimy ułatwiać sobie życie, także w grach. Wiele znakomitych usprawnień i wynalazków dokonanych zostało przecież przez leniuchów. Producenci gier często narzucają bardzo trudne warunki, wymagające wielu godzin praktyki, by choć przybliżyć się do rozwiązania. Czasem odnoszę wrażenie, że niektóre gry są dla ośmiornic i dlatego, podobnie jak wielu innych użytkowników domowych komputerów, zanim siądę do gry, próbuję ją zmienić. Zwyczaj ten, pozornie naganny, prowadzi do lepszego poznania maszyny i języków programowania, a szczególnie kodu maszynowego, czyli asemblera.

Dziś przedstawiam wyniki pracy ostatnich dwóch tygodni. Wybór gier jest właściwie przypadkowy, uzależniony od kolejności, w jakiej do mnie docierały lub od zamówienia dzieci, zmęczonych ciągłym wracaniem do początku gry.

Poniższe poprawki proponuję wprowadzić przy pomocy programu COPY COPY. Metoda ta, choć pracochłonna, jest najbardziej efektywna. Należy skopiować segmenty poprzedzające ten, do którego wprowadzamy poprawki, a następnie wczytać interesujący nas segment pod właściwy adres. Realizujemy to poleceniem LOAD AT ... (tu wpisujemy adres), pamiętając o odjęciu 17, jeżeli segment poprzedzony jest nagłówkiem. Nagłówek zawiera m.in. informację o początkowym adresie, od którego program będzie ładowany do pamięci komputera. W programie COPY COPY adres ten jest zapisany w kolumnie "start". Wartości adresu początkowego dla segmentu bez nagłówka trzeba odczytać z programu ładującego. Najczęściej można go znaleźć pod REM w programie BASIC lub w krótkim segmencie z nagłówkiem, bezpośrednio po programie BASIC. Powyższa operacja odszukania adresu startu jest pomocna, gdyż często możemy spotkać programy w różnych wersjach i adresy, które podaję dalej, mogą być niewłaściwe.

Wracamy do poprawek. Mamy wgrany właściwy segment pod odpowiedni adres. W COPY COPY klawisz O oznacza rozkaz POKE, więc naciskamy O i wpisujemy np. 37263,0. Następnie nagrywamy zmieniony segment na taśmę. Kopiujemy dalsze segmenty (jeżeli takie pozostały) i już mamy nagraną na taśmie zmienioną wersję programu.

Pierwszym programem, który będziemy zmieniać, jest zręcznościowy "HIGHWAY ENCOUNTER". Celem gry jest przeprowadzenie LASERTRON'u (mały ostrosłup, popychany przez naszą drużynę) do końca pełnej niebezpieczeństw drogi, gdzie LASERTRON zostanie uaktywniony i zniszczy statek nieprzyjaciela. Utrudnieniem gry jest limit czasu, w jakim musimy wykonać zadanie.

Wprowadzamy poprawki w trzecim segmencie z nagłówkiem (po BASIC i segmencie zawierającym tytułowy ekran). Wpisujemy POKE 37263,0, który usuwa sprawdzanie, czy limit został przekroczony. Powyższa zmiana wystarcza do skończenia gry po kilku próbach. Dla graczy o "sztywnych" palcach (do których zalicza się niżej podpisany) za miesiąc podam poprawki umożliwiające bezkarne spotkania z wszystkimi niebezpieczeństwami. Gra staje się łatwa, ale zupełnie nieciekawa, dlatego przedstawię je później.

"GIFT FROM THE GODS" – gra przygodowo-zręcznościowa, w której nasz bohater, Orestes, podczas poruszania się, a szczególnie podczas starć traci cenną energię. Poprawki, które przedstawię, umożliwiają odzyskanie utraconej siły. Program ma identyczną strukturę jak poprzedni i tak samo poprawiamy trzeci segment. Wpisujemy POKE 57402,0. Teraz wystarczy w czasie gry zatrzymać się na chwilę i powoli stan energii powraca do 100%. Efektem działania alternatywnej poprawki POKE 57408,66 jest prawie natychmiastowy wzrost poziomu energii.

"THREE WEEKS IN PARADISE" – ciekawa i efektowna graficznie gra przygodowo-zręcznościowa firmy MIKRO-GEN. Podpowiedzi do jej rozwiązania szukajcie na 27 stronie. Teraz tylko POKE 50066,0, który daje "nieśmiertelność". Poprawkę wprowadzamy do trzeciego segmentu (po BASIC i ekranie), który ma długość 38582 bajtów, zaś adresem początkowym jest 26490.


"CHUCKIE EGG 2" – struktura jak w pierwszych dwóch programach, w trzecim segmencie POKE 35453,0 daje "nieskończone życie".

"BOULDER DASH!" – ulubiona gra moich dzieci! Zmieniamy ostatni segment o długości 23346 bajtów, ładowany od adresu 24064. Piszemy POKE 26011,0, POKE 26012,0, POKE 26013,0 i w czasie gry nie ma upływu "czasu". "Nieskończone życie" uzyskujemy przez POKE 31007,0, POKE 31008,0, POKE 31009,0.

Opisane wyżej gry i poprawki do nich dotyczą komputera ZX Spectrum. Zapraszam do wspólnej zabawy i czekam na uwagi i listy o Waszych osiągnięciach w grach i przy ich zmienianiu.

GRZEGORZ CZAPKIEWICZ

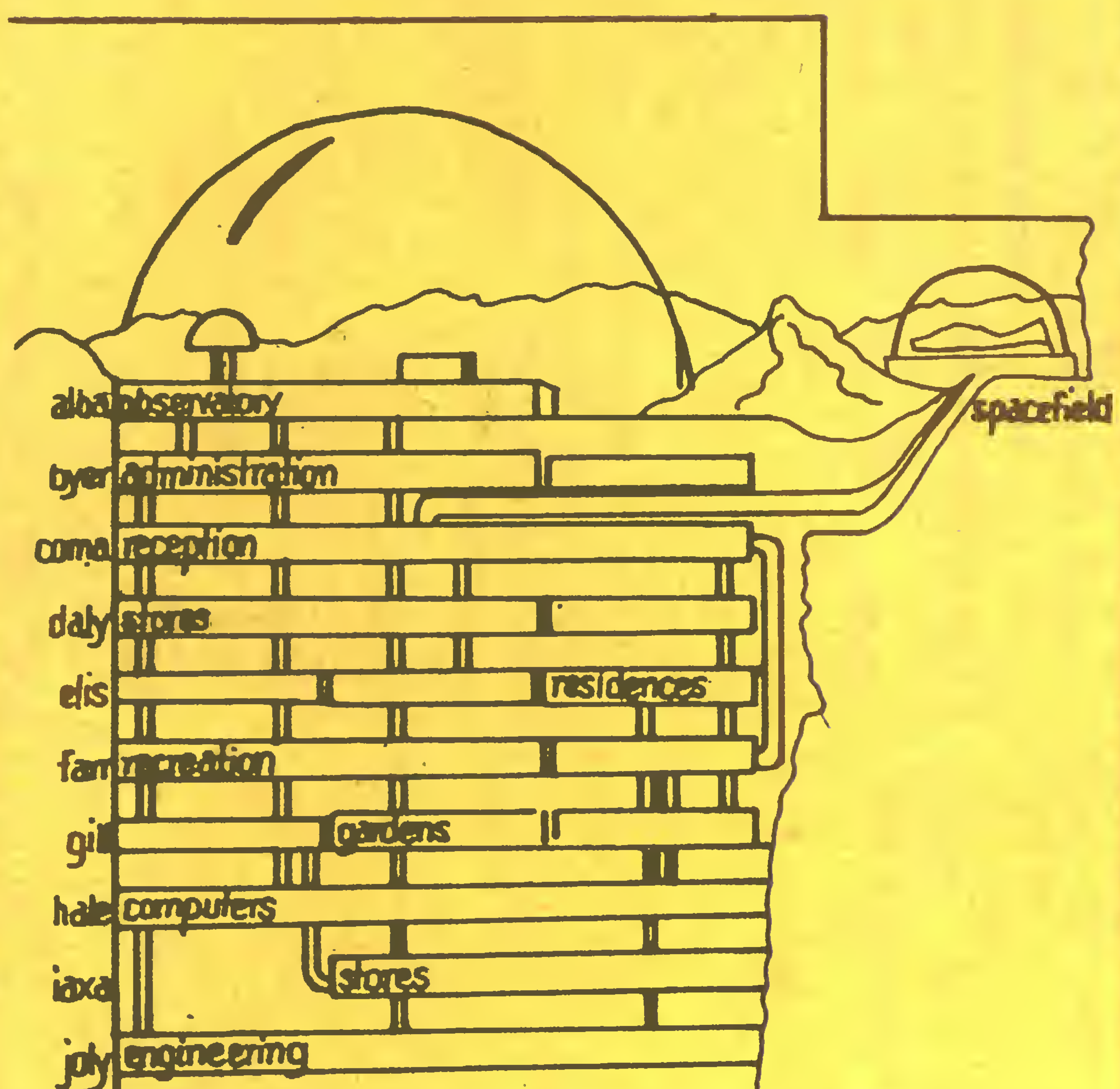
Na podstawie miesięcznika „CRASH"



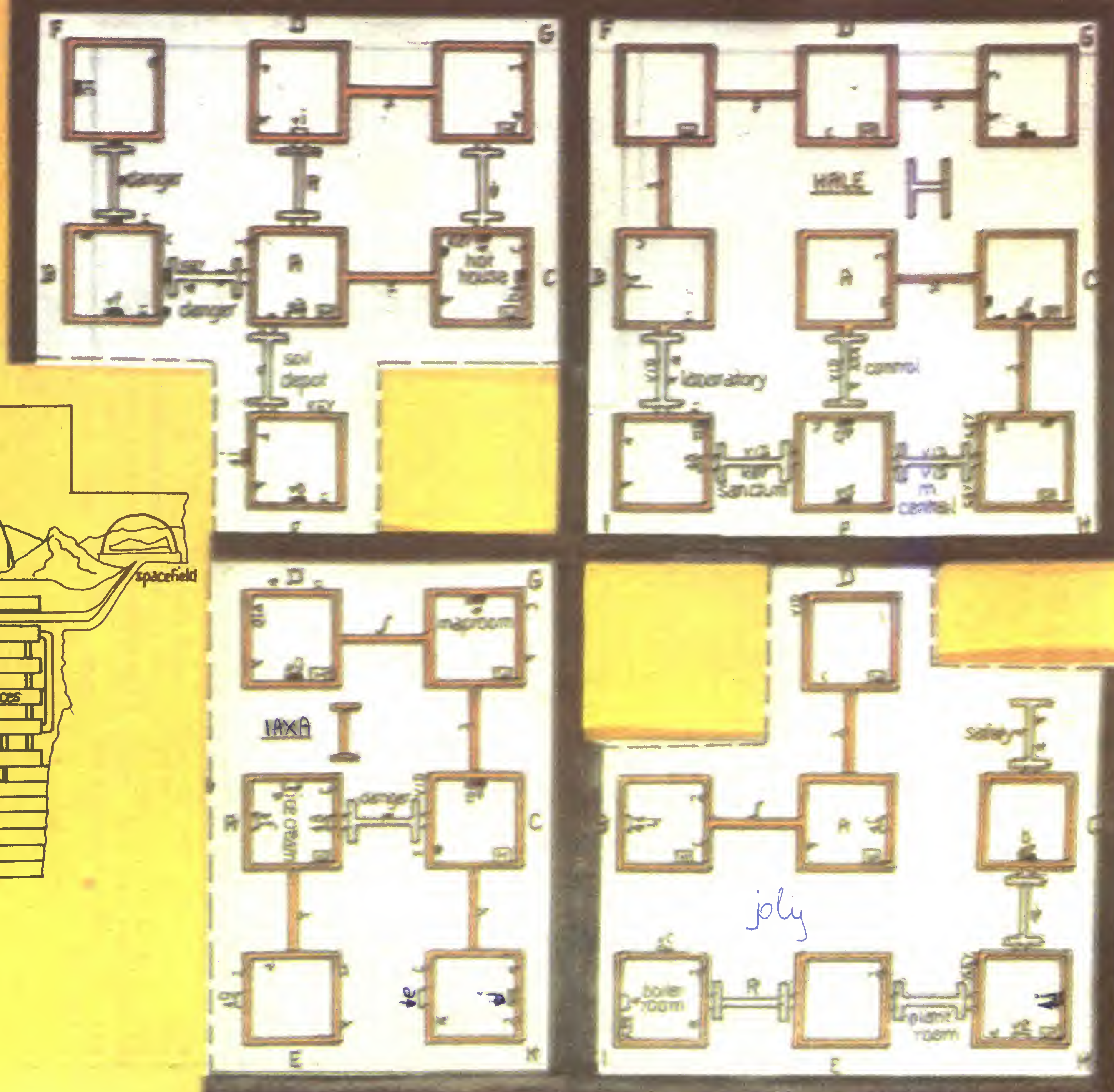
factor – charge unit
warden – door
refuse – herald
vidtex – key unit
sept – locker
warlord – supply unit

R – restricted passage
→ – one way passage
A-I – sector
a-j level tube goes to or comes from

MARSPORT



MARUSZEWSKA



DRAGONTORC [3]

Kłaniam się nisko wszystkim wytrwałym. Nie tracąc cennego miejsca (doskonale słyszę te pełne ojcowskiej troski głosy: "Streszczajcie się, Sufler. Szepcz trochę mniej rozwiekle."). Posłusznie i od razu przechodzę do sedna, czyli do trzeciego już odcinka przygód w labiryntach gry DRAGONTORC.

Mam nadzieję, że udało Wam się dotrzeć do Wiedźmich Ostępów (WITCHWOOD). Schwytajcie więc nietoperza, zerwijcie zioło i wyłówcie ze stawu żabę. Wrzućcie to wszystko do kotła. Weźcie magiczne berło, które się zeń wyłoni i z jego pomocą przesuńcie znajdującą się w sąsiedztwie kamienną płytę. Znajdziecie pod nią pergamin z zaklęciem i brylant, w zamian za który Elf da Wam brakującą połowę wizerunku Księżyca. Złóżcie obie połowy – powstanie z nich Księżycowy Klucz (MOONKEY). Dotknijcie berłem pląsającego miecza, a gdy ten się uspokoi, zetnijcie nim rosnące w pobliżu drzewko. Powstanie miotła (BRUSH), którą można pozamiatać liście. Odsłonicie w ten sposób duży głaz i pergamin z wiadomością. Dotknijcie głazu Księżycowym Kluczem. Zanim wyruszycie w dalszą drogę (a nie zapomnijcie zabrać ze sobą Klucza i miecza) odszukajcie nietkniętą dotąd kamienną płytę i wydobądźcie spod niej zaklęcie LIGHT. Teraz już możecie udać się do Ostatniego Sanktuarium (LAST SANCTUARY). Droga doń wiedzie przez odnaleziony w lesie wirujący sześcian.

Pozbądźcie się goblinów przy pomocy MISSILE. Przed atakami skrzydlatego demona ochroni Was czar MANTLE. Jeżeli dotkniecie obłożonego klątwą pergaminu, znajdziecie się w komnacie bez drzwi. Użyjcie czaru LEYROD, by się z niej wydostać. Przy pomocy miecza zdejmijcie klątwę z pergaminu. Uzyskacie wiadomość. Wymieniony jest w niej Słoneczny Klucz (SUNKEY). Ma go jeden z goblinów. W posiadaniu innego jest zielony kluczyk, który otwiera trzy skrzynie. W niektórych panują ciemności – przyświećcie sobie czarem LIGHT. Miecz pomoże Wam zdjąć klątwę z berła-pułapki, a także zamienić pająki w kilka pergaminów z kolejnymi zaklęciami. Spróbujcie odczytać i zrozumieć znalezione przy berle i w zielonej skrzyni wiadomości. Aby je w pełni wykorzystać, musicie zniszczyć skrzydlatego demona i odebrać mu czar LOCATE.

Idźcie "tam, gdzie jest najciemniej". Użyjcie czaru LIGHT, a potem przy pomocy berła przeszukajcie komnatę. Na jednej ze ścian znajdziecie klucz. Przeszukajcie również katafalk (tylko ostrożnie). Klucz otwiera zamknięte drzwi, znajdujące się w tej części Sanktuarium, w której przebywały gobliny. Otwórzcie skrzynię. Wewnątrz niej jest następny klucz. Weźcie go, zostawiając ten, dzięki któremu tu weszliście. Teraz już ostatnie zamknięte drzwi stoją przed Wami otworem. Za nimi znajdziecie wejście do dwóch komnat. W jednej z nich znajduje się niewidzialna skrzynia – tu przydaje się czar LOCATE. W skrzyni leży kolejny klucz – tym razem niezwykły, zaczarowany – zawiera wiadomość (dość enigmatyczną, przyznaję) o tym, gdzie należy szukać czterech koron, a także w jakiej kolejności je niszczyć. Druga komnata zawiera wyjście z podziemi – aby je odkryć, trzeba odwołać się do pomocy Słońca oraz Księżyca. Opuszczając Ostatnie Sanktuarium należy zabrać Słoneczny Klucz (SUNKEY), a także ten znaleziony jako ostatni, zaczarowany.

Teraz wróćcie do Katakumb Lokrisa (VAUILTS OF LOCRIS). Słoneczny Klucz otwiera jedyne zamknięte ze znajdujących się tam drzwi. Za nimi, w ciemnej komnacie, ukryty jest czar UNDO. Znajdźcie go i idźcie dalej. Omińcie ogniste kule i rozprawcie się z upiorem. Portal, prowadzący do następnego pomieszczenia, jest zaklęty. Aby bezpiecznie przezeń przejść musicie "odczynić" klątwę czarem UNDO. Za portalem znajdziecie pierwszą z czterech koron. Ponownie użyjcie UNDO, by móc ją zabrać. Przy pomocy czaru RETURN wróćcie do Świątyni Halgora (HALGOR'S SANCTUARY) i zniszczcie koronę, kładąc ją na kowadle.

Waszym następnym zadaniem jest odszukanie miejsca, zwanego Wzgórzami Snów (DREAMDOWNS). Pobyt tam rozpocznijcie od uporania się z krasnoludami (Elf Wam w tym pomoże). Potem złapcie wiewiórkę i każcie jej przeszukać liście. Zapalcie znalezioną pochodnię od ogniska w lesie. Dajcie łuk Elfowi, a w zamian otrzymacie orzech. Wewnątrz niego (znowu posłużcie się wiewiórką) jest klucz. Teraz razem z Elfem idźcie do pobliskich ruin (RUINS OF CANTII). Otwórzcie skrzynię i wyjmijcie z niej symbol węża. Dzięki niemu możecie przesunąć kamienną płytę w jednej z pobliskich komnat i zdobyć zaklęcie DEMON. Wróćcie do lasu i w taki sam sposób przesuńcie płytę, która znajduje się tuż obok kamiennego kręgu. Odsłoni się tunel. Zejdźcie nim, zabierając symbol węża i płonącą pochodnię. Potraktujcie ogniem "wytwórnię" węży, umieszczoną na ścianie pokoju przyległego do tego, przez który tu weszliście. Otwórzcie wszystkie zamknięte drzwi, dotykając ich symbolem węża. W ten sam sposób otwórzcie skrzynię. Możecie zostać przy tym zaatakowani przez węże – ochroni Was przed nimi położona obok, na podłodze pochodnia. Wewnątrz skrzyni jest jeszcze jeden symbol węża. Połączcie go z tym, który już macie. Pojawi się symbol korony. Przejdźcie przez komnatę, strzeżoną przez ogniste kule i latające topory (pomogą Wam w tym zaklęcia SLOW lub UNSEEN). Przeszukajcie katafalk przy pomocy symbolu korony, a znajdziecie rzeczywistą koronę i wiadomość. W tym samym pomieszczeniu, w którym stoi katafalk, jest również wyjście na zewnątrz. Wróćcie do Świątyni Halgora i zniszczcie koronę.

Na tym koniec (na razie). Przez najbliższy miesiąc postarajcie się dotrzeć do Trollstones. Życzę dobrej zabawy.

SUFLER

PORT

stkim należy myśleć, starać się wydedukować lub zgadnąć, co jest kluczem do zagradzających drogę drzwi, jakie przedmioty należy połączyć, aby otrzymać rzecz bardziej przydatną, którędy pójść, aby nie narażać się na niebezpieczeństwo. A to wszystko wiąże się z korzystaniem we właściwy sposób z umieszczonych w ścianach nisz. W tych, które są nazwane "Supply", można znaleźć wszystkie niezbędne do postępów w grze przedmioty. Nisze o nazwie "Locker" służą do ich bezpiecznego przechowywania. "Key" są to nisze-zamki najbliższych zamkniętych drzwi. Należy włożyć do nich klucz, czyli odpowiedni przedmiot. "Refuse" służą do pozbywania się rzeczy już bezużytecznych, zaś nisze "Charge" uzupełniają energię posiadanej przez Ciebie broni. I wreszcie nisze o nazwie "Factor" – przy ich pomocy można tworzyć z kilku przedmiotów rzeczy zupełnie nowe, dotychczas nie istniejące, bez których dalszy postęp w grze jest niemożliwy.

Główna część akcji rozgrywa się na górnej połowie ekranu. Dolna pełni funkcje informacyjne – a więc określa miejsce, w którym znajduje się komandor Marsh, pokazuje jego "kieszeń", czyli niesione przedmioty, wskazuje stopień rozładowania broni. Ponadto jest tam kompas oraz miejsce, na którym w trakcie gry wyświetlane są informacje i podpowiedzi.

Pole gry w „Marsporcie" jest podobno równe powierzchni "Tir Na Nog" i "Dun Darach" razem wziętych. Jeśli to prawda, to rzeczywiście jest gdzie prowadzić poszukiwania. W odróżnieniu jednak od tamtych gier w "Marsporcie" akcja nie rozgrywa się na jednej płaszczyźnie – miasto przypomina wielopoziomową wieżę. Poszczególne poziomy, a każdy został zaprojektowany do pełnienia innych funkcji, połączone są szeregiem wind, stających niestety tylko tam, gdzie im się podoba, a na dodatek nie zawsze dających się używać do podróży w obie strony.

W sumie to znakomita gra, chyba lepsza niż wymienione wyżej poprzedniczki. Jest tylko jedno "ale" – aby ją ukończyć, trzeba poświęcić ładnych parę tygodni. Podejrzewam, że nie robiąc przerw.

DAROSŁAW J. TORUŃ

Trzy tygodnie w raju

Rodzina Weeksów wybrała się na wakacje na piękną, tropikalną wyspę. Niestety prześladuje ich pech. Wyspę zamieszkuje plemię kanibali, a Wilma i Herbert znaleźli się w ich jadłospisie. Zadaniem Willy'ego jest uwolnienie rodzinki z łap (a może szczęk?) ponurych ludożerców, zbudowanie tratwy i ucieczka z wyspy.

Tak przedstawia się legenda najnowszej przygodowo-zręcznościowej gry firmy Mikro-Gen "THREE WEEKS IN PARADISE" (tytuł jest grą słów). Jej schemat i sposób rozwiązania jest taki, jak w znanych nam już "EVERYONE'S A WALLY" czy "HERBERT ON THE RUN". Grający, rozwiązując kolejne zagadki, prowadzą zagubionego bohatera do szczęśliwego finału. Każde prawidłowe rozwiązanie nagradzane jest melodyjką i dobudowaniem nowego elementu tratwy, która umożliwi ucieczkę. Rysunek tratwy pojawia się w dolnej części ekranu, między dwoma kościotrupami (symbol tego, co pozostanie z Wilmy i Herberta, jeżeli Willy zawiedzie) oraz czterema trupimi czaszkami symbolizującymi los Willy'ego po każdym błędzie, stanowiącymi ponadto informację, ile śmiertelnych w skutkach błędów możemy jeszcze popełnić. Ponadto lewa dolna część ekranu zawiera informacje o niesionych przedmiotach.

Sterujemy ruchami Willy'ego przy pomocy manetki (ang. joystick) typu Kempston lub z klawiatury komputera. Dowolna para klawiszy z rzędu Q-P oznacza ruch w prawo lub w lewo, dolny rząd daje nam podskok, 1 i 2 symbolizują "kieszenie", do których zbieramy potrzebne przedmioty (podniesienie lub porzucenie), zaś rząd klawiszy od A do ENTER ma dwojakie znaczenie: przejście przez bramkę albo użycie jednego lub kilku przedmiotów (np. wyrwanie ciernia szczypcami). Ponadto możemy klawiszem 5 wyłączyć miłą skądinąd muzyczkę, która jednak po pewnym czasie staje się denerwująca. Gra nie ma na szczęście limitu czasu, widocznie ludożercy nie są aż tak głodni.

Grafika jest mocną stroną gry i po załadowaniu zaczynamy razem z Willym zwiedzanie wyspy. Gdy przestaniemy zachwycać się pięknem tropikalnxch lasów, plaży czy podwodnych pejzaży zauważymy, że biedny Herbert gotuje się już w kotle, pilnowany przez okrutne lwy, zaś Wilma, zawieszona głową w dół, czeka na następny posiłek. Zauważymy czyhające na Willy'ego niebezpieczeństwa oraz porozrzucane tu i ówdzie przedmioty.

Zaczynamy systematycznie penetrować okolicę, zaglądając pod każdy kamień, za każde drzewko czy słupek. Będziemy musieli wskoczyć do studni, a także nurkować, unikając cały czas niewinnie na pozór wyglądających pszczół, motyli czy też nietoperzy lub ponurego, patrolującego dżunglę Indianina, który zdaje się nie darzyć Willy'ego sympatią. I cały czas musimy rozwiązywać niedorzeczne zagadki i odpowiadać na dziwne pytania.

Dlaczego krokodyl smutnieje i traci animusz na widok torebki Wilmy (WILMA'S HANDBAG)? Jak naostrzyć tępą siekierę (BLUNT AXE)? Jak napełnić pustą manierkę naszego starego przyjaciela Billy'ego (THE EMPTY BILLY CAN)? Dlaczego lwy są takie niespokojne? Czy "klapki" są praktycznym obuwiem (FLIPFLOPS)? Dla kogo przeznaczona jest pełna jedzenia miska (BOWL OF STUFFING)? Z czego orły budują gniazda?

Odkrywamy stary, "przedpotopowy" samochód na kamiennych kołach, które niestety nie chcą się obracać i trzeba je naoliwić. Może olejem kokosowym (COCONUT)? Czym jednak rozłupać orzech? Znajdujemy butelkę i korkociąg (BOTTLE, CORKSCREW), ale tradycyjna metoda nie daje rezultatu. Może inaczej? Właśnie tak, mamy już olej, smarujemy koła i... samochód nie jedzie. Pojawia się podpowiedź: "musisz być ostrym gościem, by zrozumieć ten rebus".

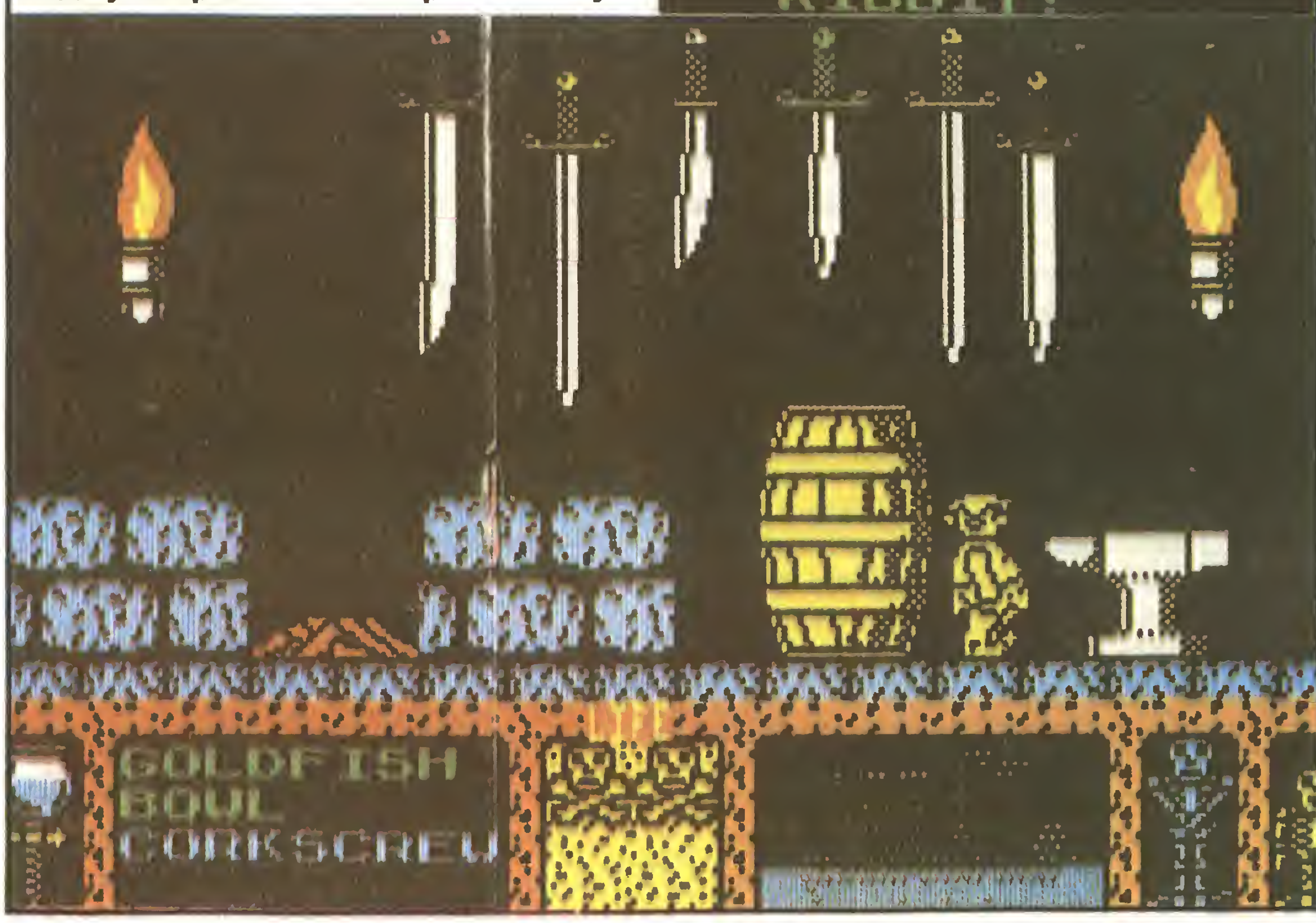
Na ruchomych piaskach spotkaliśmy tęskniącego za wodą kraba. Przechodziliśmy już obok gejzeru "Starych Przyjaciół" lecz był nieczynny. Jak go uruchomić? Może przydadzą się nasze przyzwyczajenia z łazienki? Ten wiszący łańcuszek wydaje się znajomy ...

Wdzięczny krab oddaje nam szczypce, łagodzimy cierpienia lwa (okazuje się być łagodny z natury), ale nie możemy zgasić ognia pod kotłem, z którego przerażony Herbert wzywa naszej pomocy. Na deszcz możemy długo czekać, więc poszukajmy boga deszczu. Bożek domaga się daniny w postaci gorącego popiołu (HOT ASHES). Znajdujemy "boskie drewienka" (DEUX STICKS), mijaliśmy już w swoich wędrówkach wygasłe palenisko... Jednak patyczki nie chcą się palić. Potrzebny jest miech (BELLOWS).

Bożek tańczy, zaczyna padać deszcz, biją pioruny... Może dmuchnąć w chmurę? Tylko czym? No właśnie. Pech jednak nie opuszcza Willy'ego. Piorun trafił w antenę telewizyjną domek







spłonął, a chmura nie dotarła do Herberta. Może uda się uratować ze zgliszczy choć pustą muszlę (EMPTY SEA SHELL)?

Hurra!!! Wielka radość. Czasem kropla wystarczy, by ugasić ogień. Pozostaje Wilma, mamy już ostrą siekierę (SHARP AXE), ale jak pokonać Szamana? Gdyby tak mieć łuk i strzały (BOW AND ARROWS)...

Trzeba znaleźć miętę (MINT), zanieść ją do zamarzniętego lasu (FROZEN FOREST) i ... mamy dziurę (THE HOLE!) w ścianie. Postraszymy napotkaną kreaturę złotą rybką (GOLDFISH BOWL) i zabierzemy klucz (SKELETON KEY). Otwieramy właściwe drzwi i zabieramy prowiant – puszkę szpinaku (TIN OF SPINACH). Bardzo zdrowy i pomocny przy wspinaczce do orlego gniazda nad gejzerem. Jednak orzeł, zrozpaczony utratą jedynego jaja (EGG), broni dostępu do gniazda. Jajo zostało porwane przez dziwne, wygłodzone stworzenie. Jak je odebrać?

Sprawdzamy materiał użyty na budowę gniazda i...

Uratowana Wilma rzuca się nam w ramiona, tratwa zbudowana, pozostaje tylko odpłynąć.

Stopy wody pod kilem!

GRZEGORZ CZAPKIEWICZ

Program: The Artist
Wydawca: SofTechnics – Softek
Komputer: ZX Spectrum
Autor: Bo Jangeborg

Przez trzy i pół roku obecności ZX Spectrum na naszym rynku mieliśmy okazję zapoznać się z niemałą liczbą programów graficznych, to znaczy takich, które służą pomocą wszystkim pragnącym zapełniać ekran telewizora ładnymi obrazkami. Były wśród tych programów lepsze i gorsze – przekonaliśmy się o tym sami, gdyż kilka z nich dotarło również do naszego kraju. Chyba jako ostatni – THE ARTiST. W kraju, w którym powstał, reklamowany jest jako "najpotężniejszy i najbardziej wszechstronny program graficzny, jaki dotychczas został napisany dla ZX Spectrum". Sądząc z krótkiej choćby znajomości z nim, może to być prawdą. Przyjrzyjmy się więc mu nieco bliżej.

Po dość intrygującym ekranie tytułowym (jego tworzenie zajęło podobno zaledwie dwie godziny, właśnie dzięki programowi, któremu przewodzi) główny "ekran roboczy", z rozkazami umieszczonymi na wydzielonym na dole pasie, wydaje się jakby już nieco znajomy. W rzeczywistości istnieją trzy główne zestawy rozkazów, dostępne po kolei po naciśnięciu klawisza Symbol Shift. Kursor jest sterowany za pomocą klawiszy Q i S jako góra/dół oraz R i T jako lewo/prawo. Możliwe jest również posługiwanie się manetką (ang. joystick), ale tylko systemu Kempston. Szybkość ruchu kursora zależy od czasu przyciskania klawiszy. Umożliwia to tworzenie bardzo delikatnych rysunków bez konieczności wcześniejszego określania prędkości kursora. Główne klawisze kontrolujące piórko i pędzel (PEN/BRUSH) to C – postawienie punktu, x – wymazanie go i Z – określenie koloru.



To można zrobić bez wyciągania pędzla...

AR TYSTA

Wyjaśnijmy teraz rozkazy, które widnieją na dole pierwszego, głównego ekranu roboczego. Po kolei:

BRUSH – daje nam możliwość wyboru rodzaju używanego "pędzla". Jego grubość może wahać się od 1 do 8, przy czym może to być pędzel normalny lub też aerograf, służący do "spryskiwania" ekranu – zacierania konturów czy cieniowania.

BRUSH PATTERN – z pomocą tego rozkazu możemy wybrać jeden z 16 wzorów, jakie będzie zostawiał po sobie pędzel. I tutaj również możliwy jest wybór grubości pędzla, przy czym kursor zależnie od tego wyboru zmienia kształt i wielkość.

TEXT – umożliwia umieszczenie napisów na pustym ekranie lub na uprzednio wykonanym rysunku. Kształt liter można samemu zaprojektować przy pomocy służącej do tego części programu. Ponadto sub-rozkazy dają możliwość uzyskiwania na przykład tekstu negatywowego czy też tekstu ścisłego (64 znaki w linii).

VIEW – eliminuje zawierający komendy pas na dole ekranu, odsłaniając tym samym cały rysunek. Naciśnięcie jakiegokolwiek klawisza powoduje powrót do głównego menu.

MOVE – umożliwia pracę na trzech dolnych liniach, przesuwając cały ekran ku górze.

CLS – czyści ekran, ale dopiero po powtórnym potwierdzeniu. Dzięki temu zostaje zmniejszone ryzyko artystycznych samobójstw.

STORAGE – umożliwia zgrywanie na taśmie i wprowadzanie z powrotem do komputera obrazków, zaprojektowanych liter i zdefiniowanych znaków graficznych (UDG). Sub-rozkazy dają do wyboru współpracę z magnetofonem, microdrivem lub wafadrivem (w Polsce prawie zupełnie nie znanym).

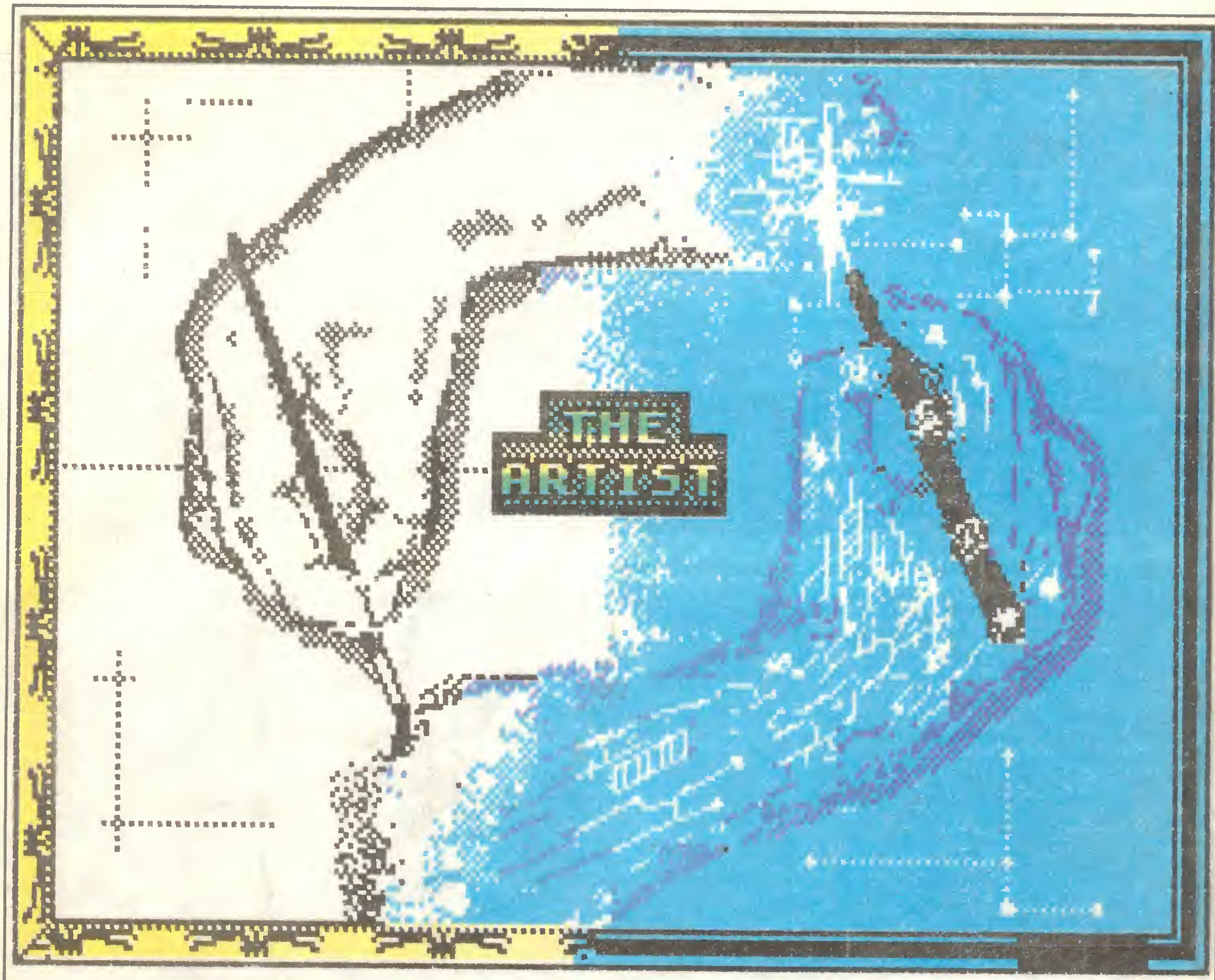
Drugi zestaw komend znajduje zastosowanie bezpośrednio w procesie rysowania.

LINE to rozkaz, przy pomocy którego rysowane są linie. Mogą tu być używane trzy odrębne techniki, przy czym każda z nich wymaga zastosowania dwóch kursorów – głównego i pomocniczego. W technice "PLOT-POINT" kursor pomocniczy jest nieruchomy i z niego biorą początek wszystkie linie, biegnące ku ruchomemu kursorowi głównemu. Technika "PLOT-MOVE" podporządkowuje kursor pomocniczy głównemu – gdy kursor główny zostanie przesunięty, przesuwa się również kursor pomocniczy. W ten sposób rysowane linie zawsze są równoległe. I wreszcie technika "PLOT-TRACE", chyba najczęściej używana. Po każdym użyciu komendy LINE przesuwa ona kursor pomocniczy do punktu, będącego ostatnią pozycją głównego kursora. W ten sposób mogą powstawać dowolne linie łamane. Kursor pomocniczy może zostać przesunięty w każdej chwili poprzez naciśnięcie klawisza SPACE.

CIRCLE – jest to prosty w użyciu sposób rysowania okręgów. Kursor pomocniczy wyznacza środek okręgu, a jego odległość od kursora głównego równa się promieniowi. Ta metoda jest dużo praktyczniejsza od używanej przez komendę CIRCLE, wprowadzoną z klawiatury ZX Spectrum, w której promień wymaga liczbowego określenia.

BOX – rysuje kwadrat, którego przekątna wyznaczana jest przez pozycje kursorów.

ARC – najpierw rysuje prowizoryczny łuk, który następnie przy pomocy sześciu klawiszy może być spłaszczany lub pogłębiany. Gdy osiągnięta



zostanie pożądana krzywizna, łuk może być utrwalony przez naciśnięcie klawisza P.

FILL – wypełnia zamknięte kształty. Może używać do tego nie tylko czerni, ale również jednego z 10 zawartych w programie wzorów lub też wzoru zaprojektowanego wcześniej przez użytkownika.

ENLARGE – powiększa ten fragment ekranu, na którym aktualnie znajduje się kursor. Wszystkie rozkazy rysujące w dalszym ciągu działają, zaś oglądany obszar może być przesuwany wzdłuż i wszerz pierwotnego obrazu.

UNDO – służy pomocą, gdy coś w trakcie rysowania poszło nie tak, jak powinno. Klawisz U wykasowuje efekty ostatniego rozkazu bądź wszystkich rozkazów, jakie zostały wydane od ostatniego naciśnięcia klawisza 0. Dzięki temu, że możliwe jest zniszczenie efektów szeregu kolejnych poleceń, można bez większych obaw przeprowadzać wszelkiego typu eksperymenty z kształtami, kolorami, cieniowaniem itd.

OVERLAY – jest to bardzo przemyślna komenda, dostarczająca komputerowemu artyście potężnego narzędzia pracy. Dzięki niej każdy fragment istniejącego obrazka może zostać wydzielony, przetworzony przy pomocy różnych poleceń, przesunięty i na powrót włączony w obraz w inny sposób. Postępowanie jest bardzo proste. Granice obszaru, który ma zostać wyłączony, oznaczane są przy pomocy techniki "PLOT-TRACE". Główny kursor powinien pozostać wewnątrz wydzielonego obszaru. Naciśnięcie klawisza 0 powoduje wypełnienie tego obszaru czernią. Po ponownym naciśnięciu klawisza OVERLAY (tzn. 3) oznaczony fragment zostanie wycięty i wyjęty z obrazka. Od tej chwili można na nim dokonywać wszelkich dostępnych operacji (na przykład odwrócenie negatywowe, odbicie lustrzane góra/dół lub lewo/prawo czy skalowanie w górę lub w dół – 10 kroków). Ponowne włączenie w ekran może również być wykonywane na kilka sposobów, a nawet w kilku miejscach.

Ostatni zestaw komend umożliwia kolorowanie rysunku, przy czym każdy z dostępnych w Spectrum kolorów może być modyfikowany funkcjami BRIGHT i FLASH. Technika jest analogiczna do poleceń wydawanych z klawiatury.

Wróćmy do pierwszego zestawu rozkazów. Znajdziemy wśród nich opcję "CHR", która zaprowadzi nas do tej części programu, która umożliwia definiowanie własnych znaków graficznych (UDG) i określanie kształtów liter. Program przewiduje możliwość zaprojektowania siedmiu zestawów znaków lub liter. Praca nad tym dokonuje się na dziewięciu dużych, niemal w całości wypełniających ekran kwadratach, z których każdy podzielony jest na mniejsze, 64-bitowe kwadraty. Jedynie niewielką część ekranu po lewej stronie zajmuje lista poleceń, wraz z umieszczonym nad nią blokiem dziewięciu znaków "naturalnej" wielkości. Własne znaki definiujemy, posługując się dostępnymi poleceniami z listy po lewej stronie – jest ich zbyt dużo, aby je tutaj omawiać, ale wystarczy odrobina praktyki, aby wszystko stało się jasne. Gdy nowy znak powstanie, wybieramy numer zestawu, któremu chcemy go przyporządkować. Numer pojawi się w lewym górnym rogu, zaś pod nim nowo zdefiniowany znak.

Różne zestawy możemy wybierać, naciskając klawisz C, a następnie numer zestawu. Gdy chcemy nagrać zestaw na taśmę, naciskamy klawisz S i wprowadzamy numer linii i numer kolumny, określające znak, od którego nagrywanie ma być rozpoczęte, a potem określamy ilość znaków. Wprowadzanie uprzednio nagranych znaków do komputera odbywa się podobnie.

Program pozwala również na "wyciąganie" znaków czy liter z głównego ekranu. Odbywa się to przez naciśnięcie klawisza K, odnalezienie ruchomym kwadratem interesującego nas miejsca i naciśnięcie klawisza X. Ponieważ ta funkcja jest odwracalna, istnieje możliwość definiowania znaków czy liter większych niż 3x3 (z pomocą komendy ENLARGE).

Przy bliższym poznaniu THE ARTIST rzeczywiście udowadnia, że ma prawo ubiegać się o miano najlepszego z programów graficznych, dostępnych dla ZX Spectrum. Duża różnorodność rozkazów i funkcji (o niektórych nawet nie wspomniałem – choćby o bardzo efektownej możliwości animowania znaków graficznych) niewątpliwie wymaga sporej praktyki. Jednak tylko niezwykle wybredny użytkownik mógłby po zapoznaniu się z nim z niezadowoleniem kręcić nosem. Oczywiście są i tacy. Ale cóż – złej baletnicy...

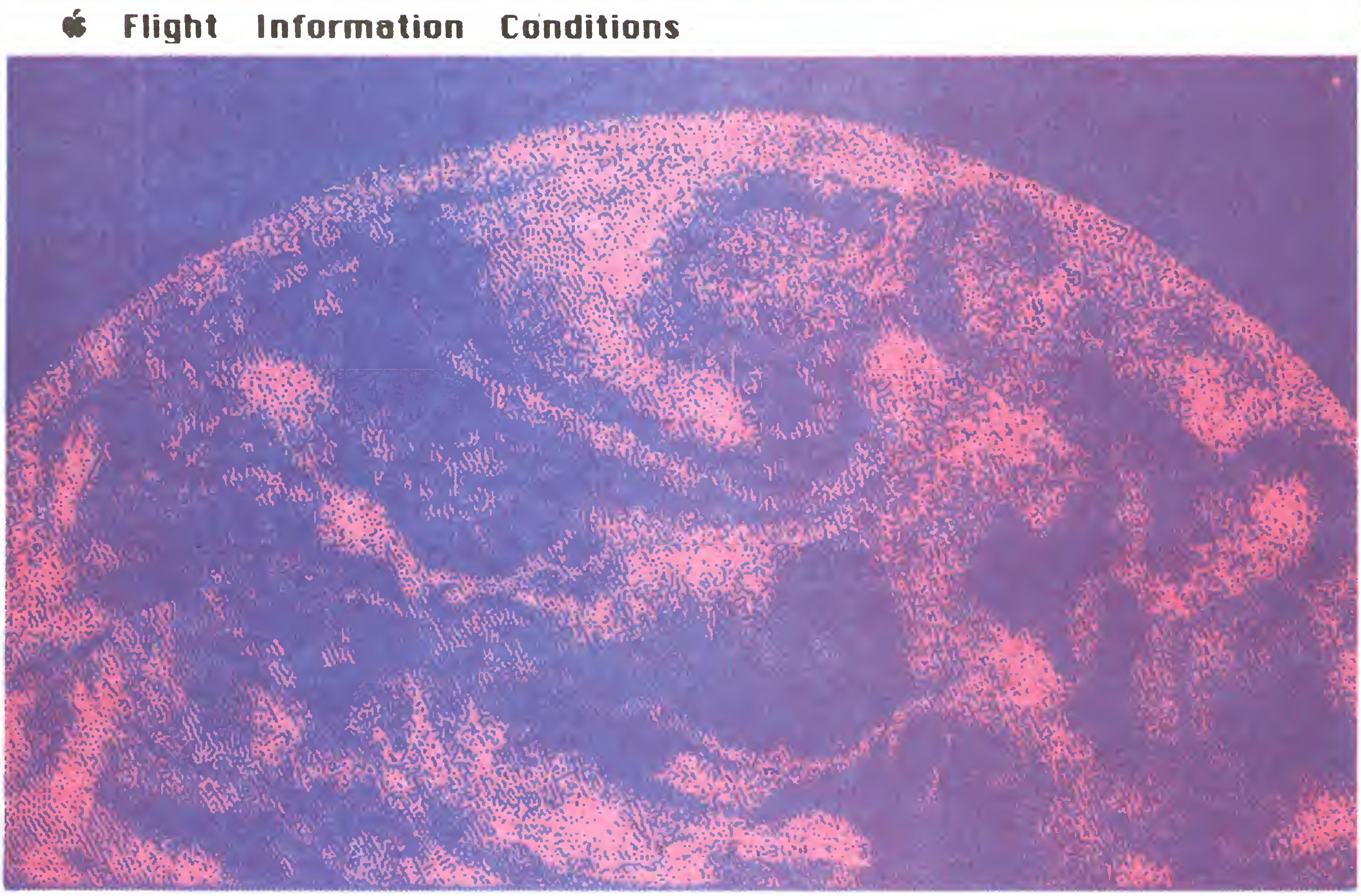
(djt)

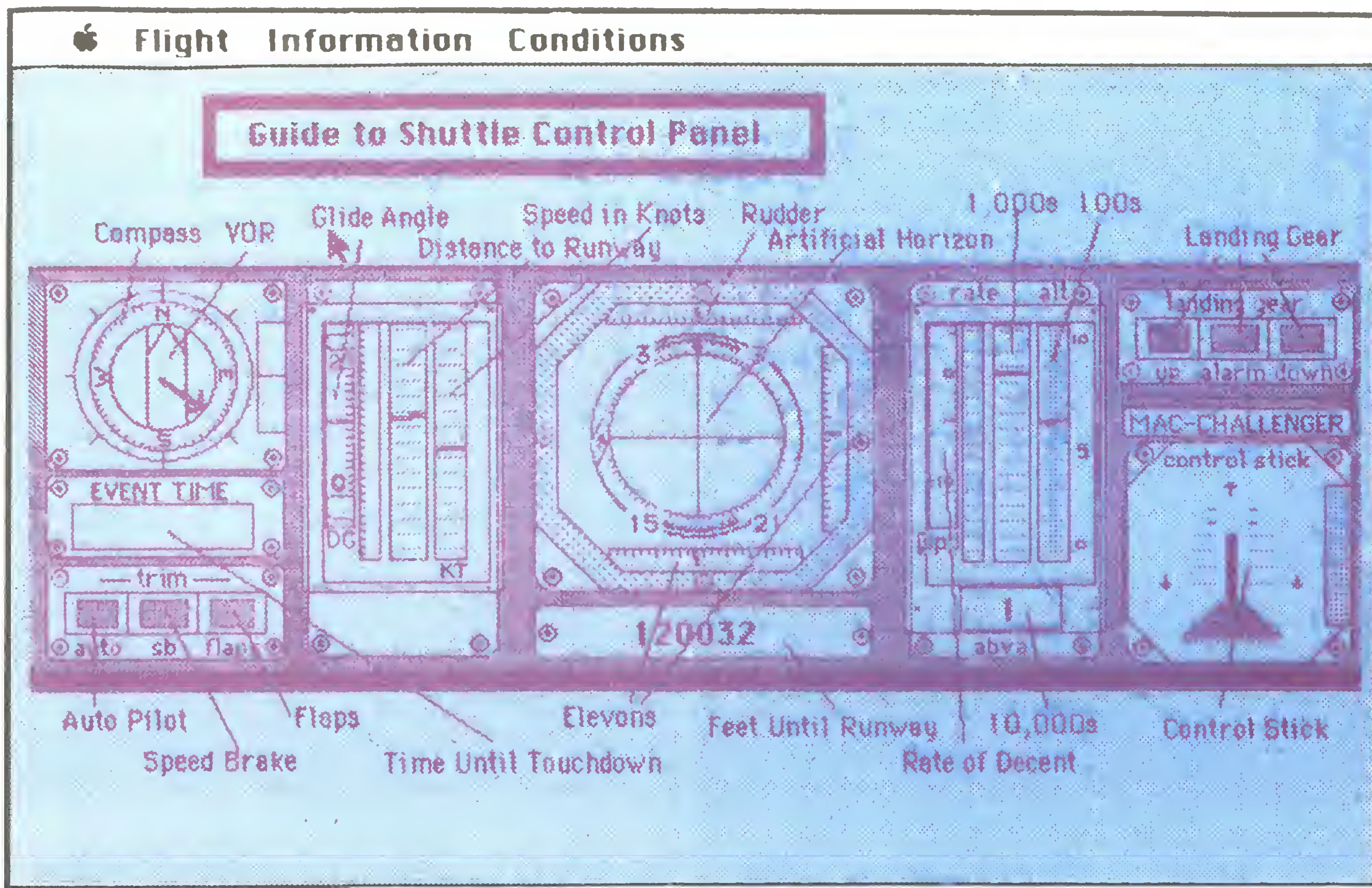


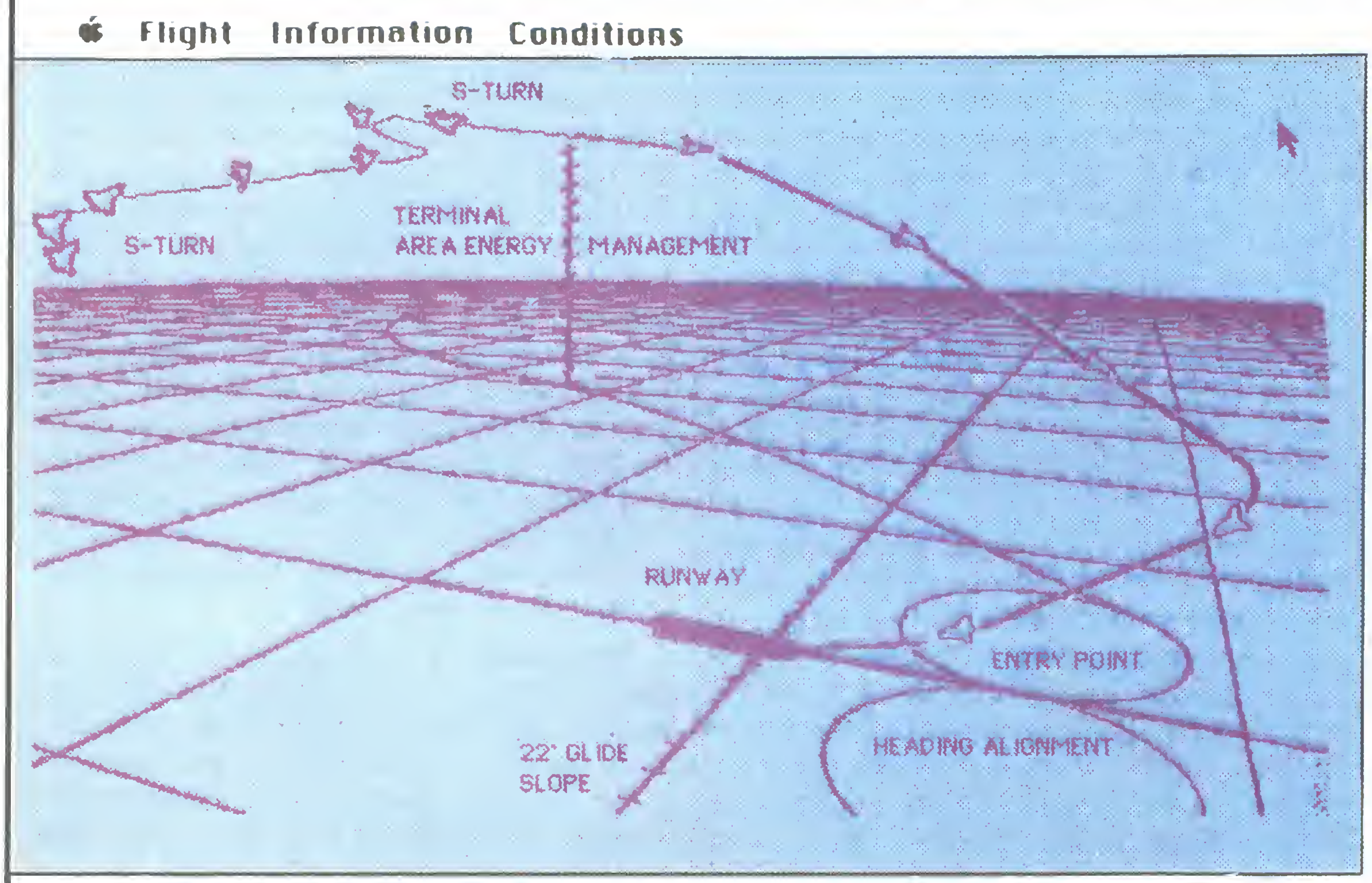
FATUM

Gdy w roku 1985 firma Aegis wypuściła na rynek nową grę dla komputera Macintosh, nikt nie przypuszczał, że w krótkim czasie stanie się ona klasykiem. Wszystkiemu, jak zwykle, winne jest życie. Dziś wystarczy powiedzieć, że gra "Mac-Challenger" jest symulacją lądowania promu kosmicznego o nazwie "Challenger". Prom ten wybuchł w dniu 28 stycznia 1986 roku w 1 minutę 12 sekund po starcie. W wypadku zginęła cała 7-osobowa załoga, w tym pierwsza całkowicie cywilna pasażerka promów, nauczycielka wybrana z 11 tysięcy chętnych do propagowania lotów kosmicznych z orbity. Firma Aegis nie zaprzestała jednak sprzedaży swego produktu – pojawiła się nawet nowa wersja gry, umożliwiająca oglądanie toru lotu. Cóż, nie ma jak reklama...

Gra "Mac-Challenger" opiera się o standardowe otoczenie programowe Macintosha: możliwości wybiera się myszą z menu, podobnie poprzez mysz steruje się promem. Symulacja jest tu nieco inna niż typowe pilotowanie samolotów (dostępne nawet na Spectrum), gdyż podczas lądowania prom kosmiczny nie korzysta z silników. Tylko umiejętności pilota pozwalają łagodnie wylądować. Prom jest więc rodzajem szybowca, tyle że o bardzo dużej prędkości. Dla mniej wprawnych przewidziano autopilota, który sam doprowadza (jak w życiu!) do pasa startowego. Jedyne co musi wtedy zrobić pilot, to wypuszczenie podwozia. Wyhamowanie prędkości zyskuje się przez gwałtowne skręty – by wylądować na pasie, trzeba





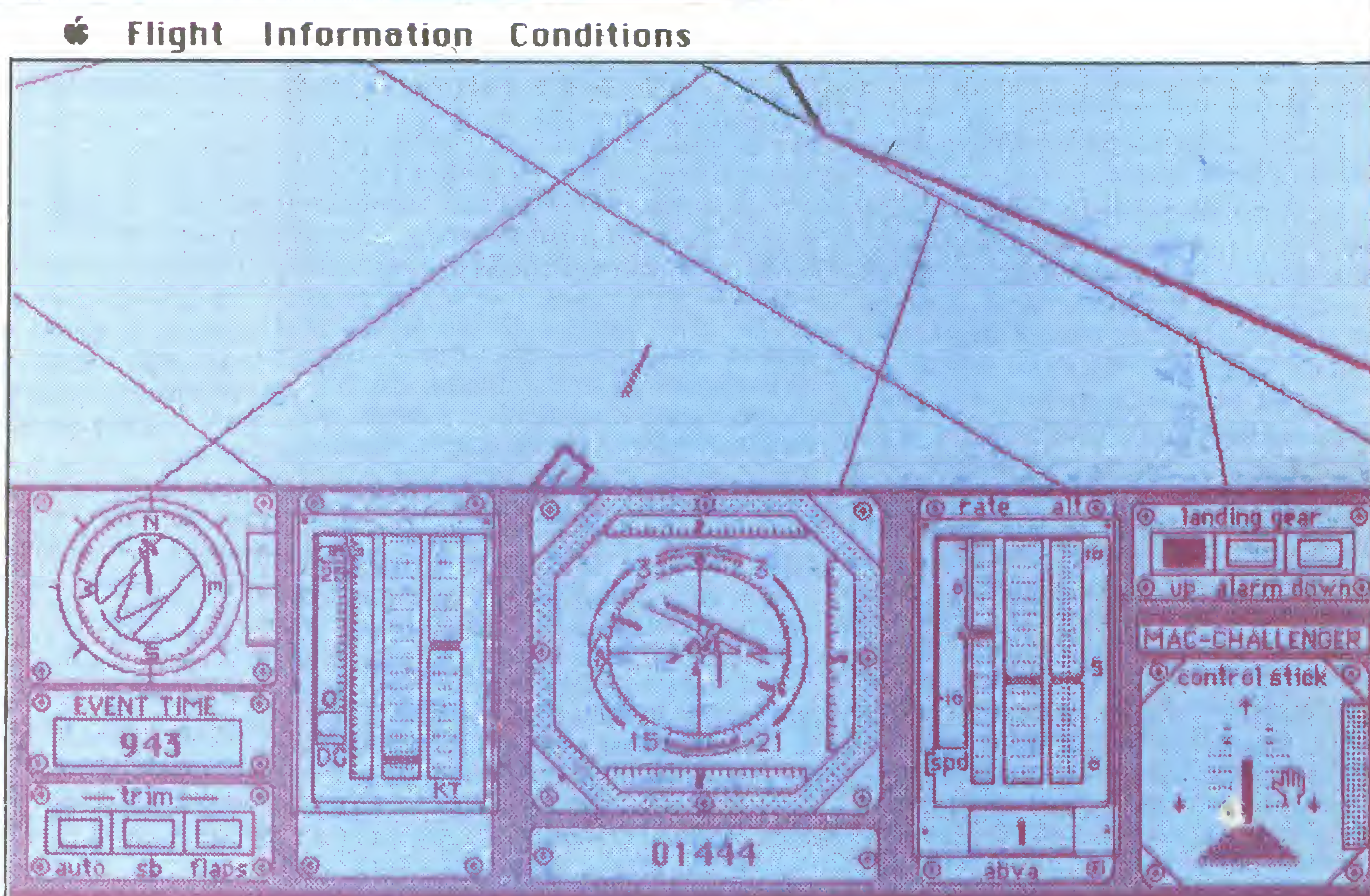


nie tylko wyhamować prędkość, ale też znaleźć się we właściwym miejscu i na właściwej wysokości.

Gra nie jest bardzo interesująca – przeciętny śmiertelnik szybko rozbija prom! Ponieważ obraz pola lądowań, pokazywany przez komputer, jest bardzo uproszczony, więc i orientowanie się w przestrzeni nastręcza sporo kłopotów. Już lepiej zażądać pełnego zachmurzenia i lecieć według przyrządów. Za to gdy rozbijemy prom, ukazuje się faksymile (niestety, nie potrafiłem go skopiować, choć inne obrazy widziane w czasie gry udało mi się wydrukować...) strony gazety z informacją o rozbiciu się promu Challenger. Sądzę, że dezintegracja prawdziwegc promu jest być może świadectwem teorii, iż zjawiska, o których często się myśli, mają wyższe prawdopodobieństwo zdarzenia. (Przypomina mi się tu znany eksperyment z pokazywaniem w angielskiej TV pewnych, niezbyt oczywistych, rysunków i następnie testowanie ich percepcji na ludziach, którzy przebywali tylko w "polu" świadomości). Jeżeli teoria ta jest prawdziwa, to tysiące ludzi, którzy rozbijali Challengera, są w jakimś sensie odpowiedzialni za katastrofę na Florydzie. A może to tylko fatum, czyli nieunikniony los?

W każdym razie po doświadczeniach z "Mac-Challengerem" nie zamierzam grać w "Balance of Power" (jest to gra, w której prowadzi się globalną politykę, używając wszystkich dostępnych środków, nawet bomb termojądrowych) – a nuż zwiększę w ten sposób prawdopodobieństwo wojny?!!!

JAKUB TATARKIEWICZ



Plansza informacyjna gry

Widok Ziemi z orbity

Informacje o tablicy przyrządów

Prawidłowy tor lotu promu kosmicznego

Ekran w czasie lotu – widać koniec pasa do lądowania oraz wieży radarowej umieszczonej na jego końcu. Kursor przyjmuje kształt dłoni, gdy zbliżamy go do drążka sterowniczego.

Najprostszy bank danych

Przepływ informacji to "wąskie gardło" naszej cywilizacji. Gromadzimy coraz większe ilości najróżniejszych danych, a równocześnie chcemy korzystać z nich w jak najwygodniejszy i najefektywniejszy sposób. "Papierowe" zbiory danych w postaci atlasów, encyklopedii, książek telefonicznych czy segregatorów z aktami zajmują wiele miejsca i nie pozwalają na szybkie i elastyczne wyszukiwanie potrzebnych informacji. Coraz większe ilości informacji gromadzimy w pamięci komputera w postaci tzw. BAZ danych. Wraz z oprogramowaniem zarządzającym obiektem informacji bazy danych tworzą BANK DANYCH.

Banki danych to jedno z podstawowych i najbardziej użytecznych zastosowań informatyki. Istnieje szereg gotowych programów do budowy takich banków: od najprostszych jak VU FILE dla ZX Spectrum do bardzo wyrafinowanych, jak dBASE III dla IBM PC i kompatybilnych. Często jednak potrzeby użytkownika mają specyficzny, indywidualny charakter i wtedy należy skonstruować własny bank danych. Stwarza to możliwość dobrego dopasowania banku do własnych potrzeb oraz doskonałą znajomość jego działania, a co za tym idzie, łatwość ewentualnej modyfikacji w przyszłości. Możemy też lepiej zrozumieć działanie innych, podobnych programów. W prostszych przypadkach stworzenia banku może podjąć się średnio zaawansowany amator w języku BASIC.

Spróbujmy zbudować prosty bank danych dla komputera domowego, rozporządzający jednak wszystkimi podstawowymi funkcjami dużych profesjonalnych banków. Program będziemy tworzyć tak, aby był on łatwy do modyfikacji i "przykrojenia" do indywidualnych potrzeb.

Objętość bazy danych jest zazwyczaj bardzo duża, co zmusza do przechowywania jej w pamięci masowej. Program został napisany pierwotnie dla komputera Commodore C-64, posiadającego stację dysków elastycznych. Aby umożliwić skorzystanie z programu zarówno użytkownikom ZX Spectrum jak i Commodore C-64 nie posiadającym stacji dysków oraz posiadaczom innych komputerów, program będziemy tworzyć dwuetapowo. W pierwszej fazie baza cała danych przechowywana będzie w pamięci operacyjnej, skąd w razie potrzeby może zostać przeniesiona na taśmę magnetofonową. W drugiej fazie baza danych może zostać umieszczona na dysku elastycznym w postaci najczęściej wykorzystywanego do tego celu zbioru o dostępie swobodnym.

Dla każdego ze zgromadzonych w pamięci obiektów zakładamy oddzielną "kartę informacyjną" zawierającą zadane informacje. W przypadku książki byłyby to np. tytuł, wydawnictwo, autor, rok wydania, objętość itp. Część z tych danych ma charakter tekstowy, pozostałe są liczbami. "Karta informacyjna" nazywana jest ZAPISEM lub z angielska REKORDEM. Najczęściej stosowane są rekordy o stałym formacie (jednolite "karty informacyjne"). Karta-rekord podzielona jest na tzw. POLA. W każdym polu zawarta jest pojedyncza dana o obiekcie. W zależności od charakteru danej, przeznaczonej do umieszczenia w określonym polu rekordu, mówimy o polach tekstowych lub numerycznych (liczbowych).

Na kartce papieru pola dla poszczególnych danych rozmieścić można swobodnie. W komputerze cały rekord zapamiętany jest zazwyczaj jako zwarta grupa komórek pamięci. W języku BASIC najprościej będzie zorganizować rekord w postaci zmiennej łańcuchowej (tekstowej) o ustalonej długości. Zmienna jest podzielona na kolejno rozmieszczone fragmenty, przydzielone poszczególnym polom. Mówimy, że długość każdego pola jest stała, tzn. na zapis każdej danej przeznaczono ustaloną ilość znaków. Jeśli potrzebna jest mniejsza ilość znaków, np. dla krótkiego tytułu, pozostała część pola pozostanie niewykorzystana. Rekord z polami o stałej długości nie wykorzystuje więc pamięci w najekonomiczniejszy sposób, ma jednak tę zaletę, że w prosty sposób można wyodrębnić poszczególne pola: znany jest numer pierwszego znaku w każdym polu, liczony od początku rekordu. Pola o stałej długości dają też stałą długość całego rekordu, co jest wygodne przy jego przechowywaniu w pamięci, zwłaszcza masowej o dostępie swobodnym (np. dysk elastyczny). Nasz przykładowy bank służyć będzie do przechowywania danych o posiadanych przez nas artykułach np. z dziedziny informatyki. Niezbędne pola to: tytuł artykułu, nazwisko autora oraz nazwa czasopisma, w którym artykuł został zamieszczony. Te pola są tekstowe. Oprócz tego potrzebna jest nam znajomość rocznika czasopisma, numeru w obrębie rocznika i strony, od której artykuł się zaczyna. Dodatkowo wprowadzimy jeszcze jedno pole tekstowe, służące do kodowania charakteru artykułu. Przyda się ono nam w przyszłości np. przy wyszukiwaniu wszystkich listingów lub wszystkich recenzji książek albo testów sprzętu komputerowego. Załóżmy, że tytuł może liczyć do 44 znaków, nazwisko autora i nazwa czasopisma – po 14 znaków. Można oczywiście poczynić inne założenia. W praktyce jednak długie nazwy trafiają się rzadko i z powodzeniem można je skrócić. Treść pól numerycznych przechowywać będziemy także w postaci tekstu. Dla rocznika wystarczą dwa znaki (dwie ostatnie cyfry roku), podobnie dla numeru w roczniku. Ponieważ trafiają się czasopisma o objętości przewyższającej 99 stron, pole numeru strony musi liczyć 3 znaki. Identyfikator charakteru artykułu może być jednoliterowy: długość pola = 1 znak Listing ="L", test sprzętu = "T", itd.). W sumie długość rekordu wyniosła 80 znaków. Informacja o strukturze rekordu, znaczeniu i charakterze pól zgromadzimy w bloku DATA, który będzie pierwszym elementem naszego programu (od linii 9999). W pierwszej linii bloku umieszczona jest ilość pól rekordu (w naszym przypadku 7), zaś w następnych trójkami: nazwa pola, długość pola i charakter zakodowany liczbą (0 – pole tekstowe, 1 – pole numeryczne).

Teraz możemy zastanowić się nad funkcjami, jakich oczekiwalibyśmy od naszego banku danych. Po pierwsze, powinien oczywiście umożliwiać zakładanie nowej bazy danych, wykorzystując wszystkie niezbędne czynności przygotowawcze i organizacyjne. Po drugie, bank danych musi umożliwiać dopisywanie nowych zapisów do bazy danych w miarę napływania np. nowych czasopism. Po trzecie, powinna istnieć możliwość przeglądania bazy danych od początku (od pierwszego zapisu) lub od dowolnie wybranego miejsca. Na życzenie wyświetlane być powinny albo kolejno wszystkie zgromadzone zapisy, albo tylko uprzednio wyselekcjonowane, spełniające określone warunki, np. wszystkie listingi wydrukowane w miesięczniku "Komputer".

W razie potrzeby musi być możliwe usunięcie zapisu z bazy danych. Sytuacja taka może zajść w dwóch przypadkach: utraty czasopisma z artykułem lub chęci przeredagowania zapisu. Stary zapis kasujemy, zaś w jego miejsce wprowadzamy nowy.

Przy użytkowaniu banku danych zachodzi konieczność wyselekcjonowania określonej grupy zapisów. Bank powinien pozwalać zarówno na selekcję zapisów na podstawie zawartości jednego lub kilku pól, jak również na kasowanie skutków selekcji, tzn. przywracanie początkowej równoprawności wszystkich zapisów w bazie danych.

Ostatnią często spotykaną czynnością jest sortowanie (porządkowanie) bazy danych w oparciu o określone kryterium. Na ogół jest to porządkowanie alfabetyczne lub w przypadku

▶ 35

Podstawowy program banku danych dla C-64

Najprostszy bank danych

```
10 REM BANK DANYCH    ROLAND WACLAWEK 1985
20 :
100 PRINT "]■■■": POKE 53280,0
110 PRINT "■ 1 - ZAKLADANIE NOWEJ BAZY DANYCH"
120 PRINT "■ 2 - WPROWADZANIE  NOWEGO  ZAPISU"
130 PRINT "■ 3 - WYSWIETLANIE ZAPISOW"
140 PRINT "■ 4 - KASOWANIE     ZAPISU"
150 PRINT "■ 5 - SELEKCJA     ZAPISOW"
160 PRINT "■ 6 - LIKWIDACJA  SELEKCJI"
170 PRINT "■ 7 - SORTOWANIE   ZAPISOW"
200 GET Z$
210 IF Z$="1" THEN GOSUB 1000: GOTO 100
220 IF Z$="2" THEN GOSUB 2000: GOTO 100
230 IF Z$="3" THEN GOSUB 3000: GOTO 100
240 IF Z$="4" THEN GOSUB 4000: GOTO 100
250 IF Z$="5" THEN GOSUB 5000: GOTO 100
260 IF Z$="6" THEN GOSUB 6000: GOTO 100
270 IF Z$="7" THEN GOSUB 7000: GOTO 100
300 GOTO 200
310 :
1000 GOSUB 9000: IF O$="N" THEN RETURN
1010 CLR: MR=400: IR=0
1020 DIM D$(MR): DIM S(MR)
1030 GOTO 100
1040 :
2000 IF IR=MR THEN PRINT"BRAK MIEJSCA": RETURN
2010 IR=IR+1:R=IR:GOSUB 9500:GOSUB 9300:RETURN
2020 :
3000 INPUT "■ NUMER PIERWSZEGO ZAPISU"; R1
3010 FOR R=R1 TO IR
3020 : IF R1<1 OR R1>IR THEN R=IR+1: GOTO 3070
3030 : IF S(R)>0 THEN GOTO 3070
3040 : GOSUB 9200: GOSUB 9400
3050 : PRINT "■ NASTEPNY ZAPIS? (T/N)"
3060 : GOSUB 9010: IF O$="N" THEN R=IR+1
3070 NEXT R: RETURN
3080 :
4000 INPUT "■ NUMER KASOWANEGO ZAPISU ";R
4010 IF R<1 OR R>IR THEN RETURN
4020 GOSUB 9200: GOSUB 9400: GOSUB 9000
4030 IF O$="N" THEN RETURN
4040 T=R: R=IR: GOSUB 9200
4050 R=T: GOSUB 9300: IR=IR-1
4060 RETURN
4070 :
5000 GOSUB 9600: GOSUB 9100: POKE 53280,5
5010 FOR R=1 TO IR
5020 : IF S(R)=1 THEN GOTO 5050
5030 : GOSUB 9200
5040 : IF MID$(R$,PP,DP)<>P$ THEN S(R)=1
5050 NEXT R
5060 RETURN
5070 :
6000 FOR I=1 TO IR: S(I)=0: NEXT I
6010 RETURN
6020 :
7000 GOSUB 9600: NK=1: POKE 53280,2
7010 WU=0: R=1: GOSUB 9200: A$=R$
7020 FOR I=1 TO IR-NK
7030 : R=I+1: GOSUB 9200: B$=R$
7040 : IF MID$(A$,PP,DP)=<MID$(B$,PP,DP) THEN 7070
7050 : R$=A$:A$=B$:B$=R$: GOSUB 9300
7060 : R$=A$: R=I: GOSUB 9300: WU=1
7070 : A$=B$
7080 NEXT I: NK=NK+1
7090 IF WU=1 THEN GOTO 7010
7100 GOSUB 6000: RETURN


READY.


9000 PRINT "■ JESTES PEWIEN ? (T/N)"
9010 GET O$: IF O$="T" OR O$="N" THEN RETURN
9020 GOTO 9010
9030 :
9100 PRINT: PRINT "▓";N$;"■";: INPUT P$
9110 IF RP=0 THEN GOTO 9150
9120 P$=RIGHT$(P$,DP)
9130 IF LEN(P$)=DP THEN RETURN
9140 P$=" "+P$: GOTO 9130
9150 IF Z$="5"AND RIGHT$(P$,1)="*"THEN DP=LEN(P$)-1
9160 P$=LEFT$(P$,DP)
9170 IF LEN(P$)=DP THEN RETURN
9180 P$=P$+" ": GOTO 9170
9190 :
9200 R$=D$(R): RETURN
9210 :
9300 D$(R)=R$: RETURN
9310 :
9400 RESTORE:READ IP:PP=1
9410 PRINT"]■              [ ZAPIS NR";R;"]■"
9420 FOR J=1 TO IP
9430 : READ N$,DP,RP: PRINT: PRINT"▓";N$;"■"
9440 : PRINT MID$(R$,PP,DP): PP=PP+DP
9450 NEXT J: RETURN
9460 :
9500 RESTORE:READ IP:PP=1:R$=""
9510 PRINT"]■              [ ZAPIS NR";R;"]■"
9520 FOR J=1 TO IP
9530 : READ N$,DP,RP: GOSUB 9100
9540 : R$=R$+P$: PP=PP+DP
9550 NEXT J: RETURN
9560 :
9600 PRINT "]■■■":RESTORE:READ IP
9610 FOR J=1 TO IP
9620 : READ N$,X,X: PRINT: PRINT J,"- ";N$
9630 NEXT J: RESTORE: READ IP: PP=1: DP=0
9640 GET O$:IF O$<"1" OR O$>CHR$(IP+48) THEN 9640
9650 FOR J=1 TO ASC(O$)-48
9660 : PP=PP+DP: READ N$,DP,RP
9670 NEXT J: RETURN
9680 :
9900 DATA 7
9910 DATA "TYTUL ",44,0
9920 DATA "AUTOR ",14,0
9930 DATA "CZASOPISMO  ",14,0
9940 DATA "ROCZNIK     ",2,1
9950 DATA "NUMER       ",2,1
9960 DATA "STRONA      ",3,1
9970 DATA "CHARAKTER   ",1,0

READY.
```

Podstawowy program banku danych dla ZX Spectrum

Najprostszy bank danych

```
100 CLS : PRINT ''': BORDER Ø
110 PRINT '"1 -ZAKLADANIE NOWEJ BAZY DANYCH"
120 PRINT '"2 -WPROWADZANIE  NOWEGO  ZAPISU"
130 PRINT '"3 -WYSWIETLANIE ZAPISOW"
140 PRINT '"4 -KASOWANIE     ZAPISU"
150 PRINT '"5 -SELEKCJA     ZAPISOW"'
160 PRINT '"6 -KASOWANIE   SELEKCJI"
170 PRINT '"7 -SORTOWANIE    ZAPISOW"
200 LET Z$=INKEY$
210 IF Z$="1" THEN GO SUB 1000: GO TO 100
220 IF Z$="2" THEN GO SUB 2000: GO TO 100
230 IF Z$="3" THEN GO SUB 3000: GO TO 100
240 IF Z$="4" THEN GO SUB 4000: GO TO 100
250 IF Z$="5" THEN GO SUB 5000: GO TO 100
260 IF Z$="6" THEN GO SUB 6000: GO TO 100
270 IF Z$="7" THEN GO SUB 7000: GO TO 100
300 GO TO 200
310:
1000 GO SUB 9000: IF O$="N" THEN RETURN
1010 CLEAR : LET MR=400: LET IR=0
1020 RESTORE : READ IP: LET DR=0
1030 FOR I=1 TO IP
1040  READ N$,DP,RP: LET DR=DR+DP
1050 NEXT I: DIM D$(MR,DR): DIM S(MR)
1060 GO TO 100
1070:
2000 IF IR=MR THEN PRINT "BRAK MIEJSCA": RETURN
2010 LET IR=IR+1: LET R=IR: GO SUB 9500
2020 GO SUB 9300: RETURN
2030:
3000 INPUT "NUMER PIERWSZEGO ZAPISU:";R1
3010 FOR R=R1 TO IR
3020  IF R1<1 OR R1>IR THEN LET R=IR+1: GO TO 3070
3030  IF S(R)>0 THEN GO TO 3070
3040  GO SUB 9200: GO SUB 9400
3050  PRINT '"NASTEPNY ZAPIS? (T/N)"
3060  GO SUB 9010: IF O$="N" THEN LET R=IR+1
3070 NEXT R: RETURN
3080:
4000 INPUT "NUMER KASOWANEGO ZAPISU ";R
4010 IF R<0 OR R>IR THEN RETURN
4020 GO SUB 9200: GO SUB 9400: GO SUB 9000
4030 IF O$="N" THEN RETURN
4040 LET T=R: LET R=IR: GO SUB 9200
4050 LET R=T: GO SUB 9300: LET IR=IR-1
4060 RETURN
4070:
5000 GO SUB 9600: GO SUB 9100: BORDER 4
5010 FOR R=1 TO IR
5020  IF S(R)=1 THEN GO TO 5050
5030  GO SUB 9200
5040  IF R$(PP TO PP+DP-1)<>P$ THEN LET S(R)=1
5050 NEXT R
5060 RETURN
5070:
6000 FOR I=1 TO IR: LET S(I)=0: NEXT I
6010 RETURN
6020:
7000 GO SUB 9600: LET NK=1: BORDER 2
7005 LET KP=PP+DP-1
7010 LET WU=0: LET R=1: GO SUB 9200: LET A$=R$
7020 FOR I=1 TO IR-NK
7030  LET R=I+1: GO SUB 9200: LET B$=R$
7040  IF A$(PP TO KP)<=B$(PP TO KP) THEN GO TO 7070
7050  LET R$=A$: LET A$=B$: LET B$=R$: GO SUB 9300
7060  LET R$=A$: LET R=I: GO SUB 9300: LET WU=1
7070  LET A$=B$
7080 NEXT I: LET NK=NK+1
7090 IF WU=1 THEN GO TO 7010
7100 GO SUB 6000: RETURN
7110:
7120:        8000 DIM N(1) :LET N(1)=IR
9000 PRINT '"JESTES PEWIEN? (T/N)"
9010 LET O$=INKEY$: IF O$="T" OR O$="N" THEN RETURN
9020 GO TO 9010
9030:
9100 PRINT INVERSE 1;'N$;: INPUT P$: PRINT P$
9105 LET V=LEN P$
9110 IF RP=0 THEN GO TO 9150
9120 IF V>DP THEN LET P$=P$(V-DP+1 TO )
9130 IF LEN P$=DP THEN RETURN
9140 LET P$=" "+P$: GO TO 9130
9150 IF Z$="5" AND P$(V)="*" THEN LET DP=V-1
9160 IF V>DP THEN LET P$=P$( TO DP)
9170 IF LEN P$=DP THEN RETURN
9180 LET P$=P$+" ": GO TO 9170
9190:
9200 LET R$=D$(R): RETURN
9210:
9300 LET D$(R)=R$: RETURN
9310:
9400 RESTORE : READ IP: LET PP=1
9410 CLS : PRINT '"     ZAPIS NR [ ";R;" ]"
9420 FOR J=1 TO IP
9430  READ N$,DP,RP: PRINT INVERSE 1;'N$;
9440  PRINT R$(PP TO PP+DP-1): LET PP=PP+DP
9450 NEXT J: RETURN
9460:
9500 RESTORE : READ IP: LET PP=1: LET R$=""
9510 CLS : PRINT '"     ZAPIS NR [ ";R;" ]"
9520 FOR J=1 TO IP
9530  READ N$,DP,RP: GO SUB 9100
9540  LET R$=R$+P$: LET PP=PP+DP
9550 NEXT J: RETURN
9560:
9600 CLS : RESTORE : READ IP
9610 FOR J=1 TO IP
9620   READ N$,X,X: PRINT 'J;" - ";N$
9630 NEXT J: RESTORE : READ IP: LET PP=1: LET DP=0
9640 LET O$=INKEY$
9645 IF O$<"1" OR O$>CHR$ (IP+48) THEN GO TO 9640
9650 FOR J=1 TO CODE (O$)-48
9660  LET PP=PP+DP: READ N$,DP,RP
9670 NEXT J: RETURN
9680:
9900 DATA 7
9910 DATA "TYTUL ",44,0
9920 DATA "AUTOR       ",14,0
9930 DATA "CZASOPISMO ",14,0
9940 DATA "ROCZNIK     ",2,1
9950 DATA "NUMER       ",2,1
9960 DATA "STRONA      ",3,1
9970 DATA "CHARAKTER  ",1,0
```

32 ♦

pól numerycznych w kolejności rosnących wartości. Pożądana jest możliwość sortowania wielopoziomowego, np. zapisy ułożone są według czasopism, w obrębie każdego czasopisma w kolejności roczników, zaś w ramach rocznika – w porządku alfabetycznym nazwisk autorów. Chociaż język BASIC kiepsko nadaje się do programowania strukturalnego, postaramy się zorganizować nasz program w sposób przejrzysty i modularny. Zależy nam na komforcie obsługi, skorzystamy więc z techniki „menu", czyli wyświetlania na ekranie zestawów czynności pojedynczym wciśnięciem klawisza.

Najważniejsze jest główne menu, obejmujące wymienione uprzednio podstawowe funkcje systemu. Ekran jest kasowany, barwa ramki zmieniana na czarną (linia 100). Następnie wyświetlany jest zestaw możliwości. Inwersyjna litera "Q" w liniach 110-170 to symbol sterujący, specyficzny dla C-64 i powodujący przejście do początku nowej linii. W ten sposób uzyskuje się podwójne odstępy między wierszami. W innych komputerach można użyć podwójnego PRINT, w ZX Spectrum – symbolu apostrofu.

Wyświetliwszy menu program oczekuje na wybór, dokonany wciśnięciem klawisza. Po wyborze klawisza "1" – "7" następuje wywołanie odpowiedniego podprogramu, realizującego daną czynność, a następnie powrót do głównego menu. W ZX Spectrum instrukcję GET Z$ zastępujemy równoważną LET Z$ = INKEY$.

Podprogram 1000 zakłada nową bazę danych. Wybór tej czynności jest decyzją o poważnych konsekwencjach, związaną ze zniszczeniem bazy dotychczas istniejącej w pamięci. Aby upewnić się w intencjach użytkownika, podprogram 1000 korzysta z usług podprogramu 9000, zadającego pytanie i akceptującego tylko odpowiedzi "T" lub "N". Jeśli odpowiedź była negatywna ("N"), podprogram 1000 kończy pracę. Gdy użytkownik wcisnąwszy "T" potwierdził zamiar założenia nowej bazy danych, kasowana jest pamięć, a maksymalna ilość rekordów ustalana na 400 (zmienna MR). Następnie deklarowane są dwie tablice jednowymiarowe rozmiaru MR. Tablica tekstowa D$ przechowuje poszczególne rekordy bazy danych, zaś tablica liczbowa S spełnia rolę pomocniczą przy selekcji rekordów. Każdemu rekordowi odpowiada jedna komórka tablicy S. Dopóki jej zawartość jest zerem, rekord uchodzi za wybrany. Początkowo cała tablica S jest wyzerowana.

W dalszej części programu sposób przechowywania bazy danych nie będzie odgrywał żadnej roli. Jedyne możliwe operacje to odczyt i zapis rekordu o wskazanym numerze. Operacje te będą odbywały się za pośrednictwem podprogramów 9200 (odczyt) i 9300 (zapis). W obydwu przypadkach w chwili wywołania podprogramu zmienna R zawiera numer rekordu. Zmienna R$ po powrocie z podprogramu 9200 zawiera odczytany rekord. Przed wywołaniem podprogramu 9300 R$ musi zawierać kompletny rekord przeznaczony do zapisu.

Zdziwi zapewne fakt powrotu z podprogramu 1000 do menu instrukcją nie RETURN, lecz GOTO 100. Otóż użyta instrukcja CLR (W Spectrum CLEAR) niszczy także informacje zawarte na stosie powrotów z podprogramów.

Podprogram 2000 dopisuje nowy rekord do bazy danych. Najpierw bada się, czy nie przekroczono maksymalnej ilości rekordów (linia 2000). Jeśli nie, ilość rekordów w bazie zwiększana jest o 1 (zmienna IR), zaś wskaźnik R wskazuje na nowo utworzony rekord. Podprogram 9500 wprowadza w dialogu z użytkownikiem zawartości poszczególnych pól rekordu, upakowując je w nakazanej kolejności w zmiennej R$. Po skompletowaniu rekordu podprogram 9300 zapisuje go w bazie danych jako jej ostatni element.

Podprogram 9500 odczytuje najpierw ze zbioru DATA ilość pól rekordu, po czym kolejno czyta nazwę pola, jego długość i rodzaj. Wprowadzeniem zawartości poszczególnych pól zajmuje się podprogram 9100. Wyświetla on inwersyjnie nazwę pola (zmienna N$) i oczekuje na wprowadzenie zawartości (zmienna P$). Dalszy tryb postępowania zależy od tego, czy pole miało charakter numeryczny, czy tekstowy. Jeśli wprowadzony ciąg znaków był dłuższy od przewidzianej długości pola, to w polu tekstowym odcięty zostanie nadmiar znaków ze strony prawej, a w numerycznym – z lewej. Jest to logiczne: w zbyt długim tytule ostatnie znaki są znacznie mniej istotne od pierwszych, zaś wprowadzenie np. roku w postaci nie dwu- lecz czterocyfrowej wymaga eliminacji, jako nieistotnych, dwóch cyfr z lewej strony. Poza tym liczba musi być dosunięta do prawego skraju pola, co będzie miało znaczenie przy sortowaniu i selekcji. Do linii 9150 powrócimy później. Gdyby długość P$ okazała się mniejsza od zadanej długości pola, to jest ona uzupełniana spacjami lewostronnie w przypadku pól numerycznych i prawostronnie w polach tekstowych.

Podprogram 3000 wyświetla kolejne rekordy. Jeśli odpowiedzią na pytanie o pierwszy rekord jest numer rekordu nieistniejącego, nastąpi powrót do głównego menu (linia 3020). Następnie sprawdza się, czy wyświetlanie rekordu o danym numerze nie zostało zabronione wskutek selekcji. Jeśli tak, program przechodzi do kolejnego rekordu, w przeciwnym razie rekord o numerze R odczytywany jest z bazy danych przez podprogram

♦ 36

Modyfikacje w programie niezbędne dla zapisu i odczytu bazy danych z taśmy w C-64

```
30 OPEN 1,1,0,"BAZA DANYCH"
40 MR=400: DIM D$(MR): DIM S(MR)
50 INPUT#1,IR
60 FOR I=1 TO IR
70 : INPUT#1,D$(I)
80 NEXT I:CLOSE 1

180 PRINT "Q 8 - ZAPIS BAZY  NA TASME"

280 IF Z$="8" THEN GOSUB 8000: GOTO 100

8000 OPEN 1,1,1,"BAZA DANYCH"
8010 PRINT#1,IR
8020 FOR I=1 TO IR
8030 : PRINT#1,D$(I)
8040 NEXT I: CLOSE1: RETURN
```

Uzupełnienia i modyfikacje w dyskowej wersji banku danych dla C-64

```
20 RESTORE: READ IP: IR=0: DR=0
30 FOR I=1 TO IP
40 : READ N$,DP,RP: DR=DR+DP
50 NEXT I: OPEN 3,8,15: MR=700: DIM S(MR)
60 PRINT"Q CZY BAZA DANYCH ISTNIEJE? (T/N)"
70 GOSUB 9010:IF O$="N" THEN GOSUB 1010:GOTO 100
80 OPEN 1,8,8,"BAZA DANYCH"
90 R=0: GOSUB 9700: INPUT#1,IR

1000 GOSUB 9000: IF O$="N" THEN RETURN
1010 CLOSE1:OPEN 2,8,15,"S:BAZA DANYCH":CLOSE 2
1020 OPEN 1,8,8,"BAZA DANYCH,L,"+CHR$(DR)
1030 R=0: IR=0: GOSUB 9700: PRINT#1,IR
1040 RETURN

9200 GOSUB 9700: INPUT#1,R$
9210 RETURN
9220 :
9300 GOSUB 9700: PRINT#1,R$;
9310 IF Z$="7" THEN RETURN
9320 R=0: GOSUB 9700: PRINT#1,IR
9330 CLOSE 1:OPEN 1,8,8,"BAZA DANYCH"
9340 RETURN

9700 HB=INT((R+1)/256): LB=R+1-256*HB
9710 PRINT#3,"P"+CHR$(8)+CHR$(LB)+CHR$(HB)+CHR$(1)
9720 RETURN
```

Modyfikacje w programie niezbędne dla zapisu i odczytu bazy danych z taśmy w ZX Spectrum.

```
10 LOAD "BAZA1" DATA D$()
20 LOAD "BAZA2" DATA S()
30 LOAD "BAZA3" DATA N()
40 LET IR=N(1)
180 PRINT '"8 -ZAPIS BANKU NA TASME"
280 IF Z$="8" THEN GO SUB 8000: GO TO 100
8010 SAVE "BAZA1" DATA D$()
8020 SAVE "BAZA2" DATA S()
8030 SAVE "BAZA3" DATA N()
8040 RETURN
```

Najprostszy bank danych

35 ◀

9200 i wyświetlany na ekranie przez 940D0. Po wyświetleniu każdego zapisu użytkownik podejmuje decyzję czy wyświetlać zapis następny, czy też powrócić do głównego menu. Jeśli odpowiedź brzmiała "N" (linia 3060), to zmienna sterująca otrzymuje wartość o 1 przewyższającą końcową. Powoduje to przerwanie wykonania pętli w najbliższej instrukcji NEXT.

Podprogram 4000 kasuje wybrany zapis. Po podaniu numeru kasowanego rekordu rekord ten jest wyświetlany (podprogram 9400), a użytkownik proszony o potwierdzenie zamiaru kasowania (podprogram 9000). W razie potwierdzenia na miejsce rekordu kasowanego wpisywany jest ostatni rekord w bazie danych, a wskaźnik liczby rekordów IR redukowany o 1.

Podprogram 5000 jest dość ciekawy. Realizuje on funkcję selekcji zapisów na podstawie zawartości wskazanych pól. Użytkownik musi wybrać jedno z pól rekordu. Ułatwia to podprogram 9600, wyświetlający w formie menu nazwy wszystkich pól. Użytkownik naciśnięciem klawisza wybiera jedno z nich. Podprogram rozpoznaje numer pola na podstawie kodu klawisza i kolejno dodając długości pól oblicza numer pierwszego znaku wybranego pola w rekordzie. Powrót do podprogramu 5000 następuje z nazwą pola (N$), jego długością i rodzajem (DP i RP). Do przeprowadzenia selekcji jest niezbędny wzorzec, który zostanie porównany z wybranym polem każdego rekordu. Jeśli nie będzie zgodności w związanej z danym rekordem komórce tablicy S, pojawi się jedynka. Będzie to oznaczało wykluczenie danego rekordu z wyświetlania w czasie przeglądania bazy danych.

Może zajść sytuacja, gdy szukamy wszystkich artykułów pewnego autora. Wtedy wystarczy wybrać do selekcji pole nazwiska, po czym podać je w pełnym brzmieniu. Po zakończonej selekcji można wywołać operację wyświetlania rekordów i na ekranie pojawią się kolejno żądane artykuły. Bardziej złożona sytuacja wystąpi wtedy, gdy zamierzamy np. przejrzeć wszystkie artykuły w "Komputerze" z roku 1986. Selekcja powinna dokonać się na podstawie więcej niż jednego pola. Najprościej osiągnąć to, poddając bazę danych kolejno dwukrotnej selekcji: raz na podstawie nazwy czasopisma, dwa na podstawie rocznika. Za drugim razem rozpatrywane będą wyłącznie rekordy nie odrzucone w pierwszej selekcji (linia 5050). Skraca to czas selekcji w dużych zbiorach.

Często występuje sytuacja, gdy nie pamiętamy pełnej nazwy artykułu lub autora, lecz znamy jedynie pierwszy wyraz lub jedną czy kilka liter. W przypadku pól tekstowych nasz program pozwala na podanie wzorca niepełnego: jednego lub kilku znaków zakończonych symbolem gwiazdki"*". Gwiazdka sygnalizuje, że wszystkie pozostałe znaki nie będą brane pod uwagę przy selekcji. Jeśli więc w charakterze wzorca podamy np. "P*", to wybrane zostaną tytuły: "PASCAL", "PROLOG", "PROGRAMOWANIE W LOGO" i "PORTY SZEREGOWE". Jeśli podamy "PRO*", to wyselekcjonowane będą tylko tytuły: "PROLOG" i "PROGRAMOWANIE W LOGO".

Wprowadzanie wzorców niepełnych umożliwia linia 9150 podprogramu 9100. Jeśli wybrana została funkcja selekcji (Z$="5") i równocześnie ostatnim znakiem podanego łańcucha P$ jest gwiazdka, to długość pola, brana pod uwagę przy selekcji, redukowana jest do wartości odpowiadającej ilości znaków na lewo od gwiazdki.

Przeprowadzona selekcja, mimo że chwilowo pożyteczna, odcina nam dostęp do pozostałych rekordów. Selekcję można skasować wybierając "6" w trybie głównego menu. Wywołany podprogram 6000 po prostu zeruje wszystkie komórki tablicy S.

Ostatnia funkcja banku danych to porządkowanie rekordów. Wykonuje to znany już podprogram 9600. Po wprowadzeniu numeru pola barwa ramki zmienia się na zieloną, co sygnalizuje początek sortowania.

Do sortowania zastosowano metodę tzw. bąbelkową. Polega ona na porównywaniu ze sobą odpowiednich pól w parach sąsiednich rekordów. Jeśli rekordy w parze są uporządkowane właściwie, to przechodzi się do następnej pary sąsiadów. W przeciwnym razie rekordy zamieniane są miejscami. Cały proces powtarza się tak długo, aż w kolejnym przejściu nie będzie potrzeby przestawienia ani jednej pary rekordów. Będzie to oznaczało, że cały zbiór jest uporządkowany. Przed każdym przeglądaniem tablicy zerowany jest wskaźnik uporządkowania WU. Wskaźnik ten przyjmuje wartość 1 przy każdej zamianie rekordów. Jeśli więc po przejściu całego zbioru WU=0, oznacza to, że porządkowanie jest zakończone. Trzeba jeszcze wyzerować tablicę S, której komórki nie były przestawiane, i można wrócić do głównego menu.

Zastosowana metoda sortowania ma wiele wad i niezbyt dobrze nadaje się do porządkowania zbiorów w pamięci masowej. Jej zalety to prostota oraz fakt, że w razie występowania w zbiorze grupy kilku elementów o identycznym polu, w grupie tych rekordów zachowane zostanie uporządkowanie istniejące poprzednio. Innymi słowy, jeśli najpierw poddaliśmy bazę danych sortowaniu według nazwy czasopisma, to baza danych będzie podzielona na grupy rekordów według czasopism, wewnątrz każdej z tych grup obowiązywać będzie podział na roczniki, zaś w danym roczniku rekordy uporządkowane będą według wzrastającego numeru wydania. Sortowanie jest operacją czasochłonną i w przypadku dużych baz danych może trwać nawet kilka godzin.

Baza danych przechowywana w pamięci istnieć będzie tylko do wyłączenia zasilania. Trzeba pomyśleć o jej zapisie w pamięci zewnętrznej. W praktyce zaraz po uruchomieniu programu przez RUN wczytywana będzie baza danych zapisana na taśmie w poprzedniej sesji. Rezygnując z wczytywania bazy danych uruchomimy program przez RUN 100, przechodząc bezpośrednio do głównego menu. W tym ostatnim dodamy jeszcze jedną, ósmą funkcję, polegającą na zapisaniu na taśmie aktualnej bazy danych. Jako pierwsza zapisywana jest liczba rekordów IR, następnie kolejne elementy tablicy D$. Niezbędne uzupełnienia programu ukazuje listing 2a (C-64) oraz listing 2b (Spectrum). W tym ostatnim problem odczytu i zapisu danych został rozwiązany odmiennie, Spectrum bowiem pozwala zapisywać i odczytywać wyłącznie tablice i to w całości. Na taśmie zapisywana jest także tablica S, co pozwala przechować w pamięci masowej także wynik ostatniej selekcji.

Najlepszym rozwiązaniem niestety ciągle niezbyt popularnym (ach, te ceny!) jest umieszczenie bazy danych na dysku elastycznym w postaci zbiorów o dostępie bezpośrednim. Zbiór o dostępie bezpośrednim (typ REL) podzielony jest na rekordy o ustalonej długości, odpowiadające zapisom w bazie danych. W przypadku C-64 i stacji dysków elastycznych zbiór o dostępie bezpośrednim może liczyć do 720 rekordów. W razie potrzeby użycia większej bazy danych można założyć równocześnie dwa lub więcej zbiorów o bezpośrednim dostępie. Dyskowa baza danych pozwala pracować bez uciążliwego wczytywania lub zapisywania na taśmie bazy danych. Bezpośrednio po załadowaniu do pamięci i uruchomieniu programu oraz założeniu dyskietki z danymi bank danych gotów jest do pracy.

Na wstępie program pyta o istnienie bazy danych. Jeśli baza jeszcze nie istnieje, następuje przejście do głównego menu. W przeciwnym razie zbiór bazy danych jest "otwierany", czyli przygotowywany do zapisu i odczytu. W naszym przykładzie założyliśmy do 700 zapisów. Zapis o numerze 1 w bazie danych przechowywany jest na dysku jako rekord nr 2, zapis 2 jako rekord nr 3 itd. Spowodowane jest to faktem wykorzystywania dyskowego rekordu nr 1 do przechowywania aktualnej liczby rekordów IR.

Istotnej zmianie uległ podprogram 1000 (zakładanie nowej bazy danych), który kasuje starą bazę danych, oblicza całkowitą długość rekordu sumując długości poszczególnych pól i otwiera na dysku nowy zbiór typu REL o nazwie "BAZA DANYCH" i wyliczonej uprzednio długości rekordu. Zmienione zostały oczywiście także podprogramy dostępu do bazy danych 9200 i 9300. Obydwa korzystają z podprogramu 9700, który przygotowuje do zapisu lub odczytu rekord dyskowy o numerze R+1. R=0 oznacza ustawienie pierwszego rekordu, mieszczącego ilość rekordów w całej bazie danych.

Przy każdym zapisie do bazy danych związanym ze zmianą wartości IR, tzn. we wszystkich operacjach z wyjątkiem sortowania, na dysk zapisywana jest oprócz rekordu R$ także i aktualna wartość IR (linie 9320-9340).

Przedstawiony program nie ma ambicji do maksymalnego wykorzystania możliwości C-64. Został napisany z użyciem minimalnego zestawu instrukcji, aby ułatwić ewentualną adaptację dla innych komputerów. Można w nim wprowadzić szereg ulepszeń. Najbardziej opłacalna jest zmiana algorytmu sortowania na bardziej złożony, lecz efektywniejszy i powiększenie ilości dostępnych zapisów (trzeba wtedy otworzyć jeszcze jeden lub dwa zbiory, a programowi 9700 oprócz pozycjonowania powierzyć jeszcze funkcje przełączania kanałów w zależności od wartości R). Warto postarać się też o diagnostykę ewentualnych błędów dyskowych.

Program jest bardzo łatwy w adaptacji do innych zastosowań. Wystarczy zmienić tylko specyfikację postaci rekordu w liniach DATA. Ilość pól w rekordzie może wynosić 1-9, ogólna długość rekordu nie powinna przekroczyć 80 znaków (w przypadku C-64). Taka jest bowiem maksymalna możliwa długość łańcucha wprowadzanego instrukcją INPUT. Można ominąć ten problem, wczytując rekordy instrukcją GET w pętli. Wydłuża to jednak czas odczytu rekordu. Nie korzystając z dysku można pozwolić sobie na rekordy długości do 255 znaków. W tym wypadku może jednak zajść konieczność redukcji MR, gdyż baza danych może po prostu nie zmieścić się w pamięci.

W żadnym wypadku nie wolno przerywać programu klawiszem STOP lub BREAK podczas sortowania. Grozi to pomieszaniem rekordów. W przypadku C-64 można zapobiec przypadkowemu naciśnięciu klawisza BREAK instrukcją POKE 808, 239. Wprowadzając POKE 808, 225 można uniknąć przerwania programu nawet przez STOP/RESTORE.

ROLAND WACŁAWEK

Wordpisemko

No i stało się. Przyszli, postawili na biurku, podłączyli. Moja maszyna do pisania powędrowała na mały stoliczek pod okno. Wszystkie długopisy i spinacze musiałam schować do szuflady, nawet dziurkacz zmienił miejsce. Szef cieszy się, jakby dokonał cudu. Może rzeczywiście to cud? Kiedy kilka miesięcy temu zakomunikował mi, że unowocześnia biuro wprowadzając komputery, myślałam, że będzie jak zwykle: dwa pisemka, brak dewiz i spokój. A tu masz: pod hasłem usprawnienia i ułatwienia pracy swojej sekretarce (to znaczy mnie) szef zrobił z mojego biurka pulpit komandora kosmodromu. Zamiast maszyny do pisania telewizor, zamiast miejsca, na którym mogłam położyć blok i coś zapisać – pudło z przyciskami, zamiast kalendarza – drugie pudło ze szparą jak skrzynka na listy. Te przyciski to podobno klawiatura komputera, to drugie natomiast – to stacja dysków (nazwa jak z lekkoatletycznego klubu kolejarzy!).

Szef mówił, że po otrzymaniu sprzętu nowej generacji – no właśnie tego komputera – moja praca się zmieni. Że teraz to już będę nowoczesną sekretarką, która zwolniona z absorbujących prac technicznych może z powodzeniem przyjmować obowiązki wymagające inwencji i pomysłowości. Według mojego szefa to nowe urządzenie będzie pamiętać o wszystkim (tak jakbym ja o czymś zapominała!), będzie podawało wszystkie informacje na każde zawołanie i to superdokładnie. Ten komputer ma sam przygotować wszystkie materiały na naradę i przedstawić wariantowe plany. Sam zorganizuje konferencję w optymalnym, dogodnym dla wszystkich terminie i pomoże w podejmowaniu decyzji. Po prostu zastąpi trzech wicedyrektorów i pięć sekretarek.

Poczułam się niepotrzebna. Gdy napomknęłam o tym szefowi, natychmiast dowiedziałam się, że komputer dla mnie ma być przyjacielem i koleżanką. Ma zastąpić maszynę do pisania i woźną, telefon i gońca. Dzięki tej wspaniałej maszynie już nie będę potrzebowała notesu ani kalkulatora, ani kalendarza. Z szafy mają zniknąć wszystkie segregatory i skoroszyty. Niestety, szef nie powiedział, po co ja mam zostać. Chyba do parzenia kawy, bo tego najlepszy nawet mózg elektronowy nie potrafi.

Ze wszystkiego, co mówił szef, prawdą jest jedno: moja praca się zmieniła. Przede wszystkim ekran zasłania mi drzwi, więc nie widzę kto wchodzi i wychodzi. Muszę wstawać z krzesła na dzwonek telefonu, bo przewód jest za krótki, by można było ominąć aparaturę i postawić aparat gdzieś pod ręką. Wszystkie przybory, a nawet kalendarz, trzymam w szufladzie, bo na biurku nie zostało ani kawałka miejsca. Z tego powodu notatki muszę robić kładąc papier na podkładce ze starego skoroszytu opartego na kolanach.

Nie myślcie jednak, że jestem przeciw wprowadzaniu nowoczesności. Nie, wręcz przeciwnie, zawsze byłam za postępem: od razu zmieniłam zwykłą maszynę do pisania na elektryczną, z radością przyjęłam i kalkulator, i domofon. Postanowiłam więc nie dać się i teraz. Przecież do czegoś to wszystko musi służyć. Wzięłam do domu instrukcje, które były w tych samych kartonach, co komputer, i udając przed mężem i dziećmi, że mam pilną pracę, zaczęłam czytać. Jakie szczęście, że w szkole uczyłam się angielskiego! Po kilku dniach już wszystko wiedziałam, tak przynajmniej mi się wydawało.

Wtedy właśnie szef polecił mi napisać pismo do Zjednoczenia na "wordprocesorze" – to znaczy na tym moim urządzeniu. Myślałam, że pójdzie tak, jak było napisane w instrukcji: wpisujesz jak leci, poprawiasz, komputer zredaguje i koniec. Moje nadzieje, rozbudzone reklamową propagandą firmy IBM, spełzły jednak na niczym. Oczywiście szybko wpisałam te pół stroniczki (kurs maszynopisania zaliczyłam na piątkę!), ale wstyd powiedzieć, co z tego wyszło.

Wyobraźcie sobie, że dopiero teraz okazało się, iż dostałam jakiś wybrakowany egzemplarz: litery "Z" i "Y" mają zamienione klawisze! Szef powiedział, że tak powinno być, bo tego wymaga epoka komputerów. Jeżeli to jest właśnie nowoczesnością, to ja dziękuję, przeniosę się do pracy w archiwum.

W końcu musiałam pozamieniać te zety i igreki. Nie będę opisywała innych poprawek. Powiem wam tylko w tajemnicy, że wolę poprawiać zwykły rękopis, nawet napisany charakterem pisma mojego szefa, niż tekst na edytorze tekstu. Na ekranie, gdy skreślę jakiś wyraz, znika on zupełnie. Gdy okaże się, że jednak poprzednia wersja była lepsza, trzeba wpisać ją od nowa. Żeby nie ryzykować, że o czymś zapomnę, wymyśliłam sposób: zaraz po wpisaniu do komputera przepisałam sobie z ekranu na karteczkę cały tekst i dopiero wtedy przystąpiłam do poprawiania.

Wreszcie po trzech godzinach na ekranie miałam gotowe pisemko. Przepisanie go na maszynie nie stanowiło problemu. Wiecie już, że piszę szybko, a stolik z maszyną przesunęłam tak, by widzieć ekran telewizora.

Gdy dałam przygotowane z takim pietyzmem pismo do podpisu, szef aż cmoknął z zachwytu. "Tak", powiedział, "widać, że wspinamy się w górę po szczeblach nowoczesności". Potem dodał coś, przez co nie mogę spać już od dwóch dni. Po prostu zaczęłam się bać. Wprowadzenie komputera na moje biurko już jakoś przeżyłam, ale na kolejny skok cywilizacji mam za słabe nerwy.

Chcecie wiedzieć, co mnie tak przestraszyło? Otóż szef powiedział, że chce uzupełnić konfigurację naszego sprzętu. Aby lepiej odczuć dobrodziejstwa komputeryzacji, za kilka tygodni chce sprowadzić drukarkę! Jak go znam, na pewno dopnie swego.

Agnieszka



BEATA MARUSZEWSKA

6502 Wstęp do programowania Lista rozkazów

W poprzednim artykule na temat 6502, oprócz podstawowych informacji została przedstawiona architektura wewnętrzna tego mikroprocesora. Jej podstawową cechą jest prostota – nie ma ani jednego rejestru 16-bitowego, brak też wewnętrznego banku rejestrów uniwersalnych. Rysunek 1 przedstawia konfigurację rejestrów dostępnych dla programisty. Oprócz akumulatora A, dwóch rejestrów indeksowych X i Y, służących też do tymczasowego przechowywania danych, i rejestru wskaźników F, istnieje jedynie 8-bitowy wskaźnik stosu S i dwuczęściowy licznik rozkazów PCH/PCL.

Osiem bitów rejestru S ogranicza długość stosu do 256 bajtów, co uniemożliwia przesyłanie przez niego większych struktur danych. Dzięki rozbudowanym trybom adresowania nie istnieje praktycznie problem programowy, dla którego byłoby to ograniczeniem, natomiast taka organizacja zapewnia dużą szybkość wykonywania operacji ze stosem. Jest to niebagatelną zaletą 6502.

Rejestr wskaźników F jest przedstawiony na rysunku 2. Symbole oznaczają następujące wskaźniki:

N – negative – najbardziej znaczący bit wyniku ostatniej operacji,

V – overflow – 1, gdy wynik ostatniej operacji nie zmieścił się na 8 bitach i przeniesieniu C,

B – break – wskazuje na wykonanie instrukcji BRK,

D – decimal – ustawiany przez programistę wskaźnik dziesiętnego trybu pracy. Gdy D = 1, rozkazy ADC i SBC operują na liczbach w kodzie BCD i nie wymagają żadnych korekcji,

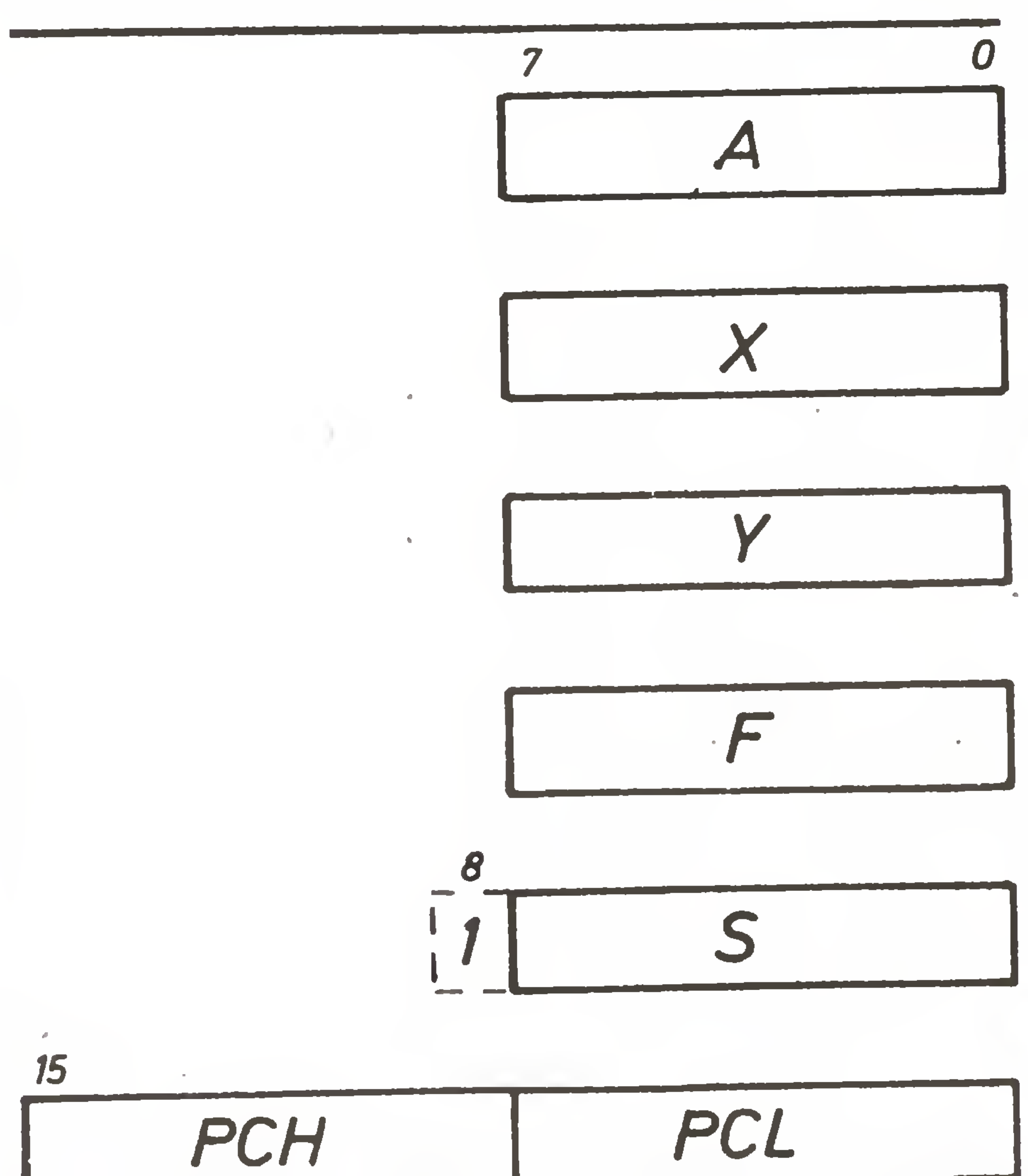
I – interrupt – ustawiana przez programistę blokada przerwań,

Z – zero – 1, gdy wynik ostatniej operacji wynosi 0,

C – carry – przeniesienie.

Bardzo ważną cechą wskaźników N i Z jest funkcjonowanie ich podczas każdej operacji arytmetycznej i logicznej oraz przy większości przesłań. Nie ma zatem potrzeby używania instrukcji, których jedynym celem będzie uruchomienie wskaźników, jak to ma miejsce w przypadku np. Z 80.

Pamięć, z którą współpracuje 6502, dzieli się na strony – każda po 256 (100_H) bajtów. W obrębie jednej strony operacje na adresach są wykonywane tylko przy użyciu mniej znaczącego bajtu licznika rozkazów PC. Na prostym fakcie większej szybkości wykonywania operacji 8-bitowych od 16-bitowych, jest oparta koncepcja szczególnego wykorzystania strony o adresie zerowym. Wszystkie rozkazy operujące na argumentach o adresach od 0 do FF_H wymagają jedy-



3. OPERACJE ARYTMETYCZNE I LOGICZNE, PRZESŁANIA DO- I Z PAMIĘCI

ADC	Add with Carry	--	69	65	6D	75	7D	--	79	61	71	NV....ZC
	(dodaj z przeniesieniem)		2	3	4	4	+4		+4	6	+5	
AND	Logical And	--	29	25	2D	35	3D	--	39	21	31	N.....Z.
	(iloczyn logiczny)		2	3	4	4	+4		+4	6	+5	
ASL	Arithmetic Shift Left	0A	--	06	0E	16	1E	--	--	--	--	N......ZC
	(przesuń w lewo)	2		5	6	6	7					
BIT	Bit test	--	--	24	2C	--	--	--	--	--	--	NV....Z.
	(testuj bity)			3	4							
CMP	Compare with A	--	C9	C5	CD	D5	DD	--	D9	C1	D1	N.....ZC
	(porównaj z A)		2	3	4	4	+4		+4	6	+5	
CPX	Compare with X	--	E0	E4	EC	--	--	--	--	--	--	N.....ZC
	(porównaj z X)		2	3	4							
CPY	Compare with Y	--	C0	C4	CC	--	--	--	--	--	--	N.....ZC
	(porównaj z Y)		2	3	4							
DEC	Decrement Memory Byte	--	--	C6	CE	D6	DE	--	--	--	--	N.....Z.
	(zmniejsz bajt w pamięci)			5	6	6	7					
EOR	Exiusive Or	--	49	45	4D	55	5D	--	59	41	51	N.....Z.
	(logiczne "albo")		2	3	4	4	+4		+4	6	+5	
INC	Increment Memory Byte	--	--	E6	EE	F6	FE	--	--	--	--	N.....Z.
	(zwiększ bajt pamięci)			5	6	6	7					
LDA	Load A register	--	A9	A5	AD	B5	BD	--	B9	A1	B1	N.....Z.
	(załaduj akumulator)		2	3	4	5	+4		+4	6	+5	
LDX	Load X register	--	A2	A6	AE	--	--	B6	BE	--	--	N.....Z.
	(załaduj rejestr X)		2	3	4			4	+4			
LDY	Load Y register	--	A0	A4	AC	B4	BC	--	--	--	--	N.....Z.
	(załaduj rejestr Y)		2	3	4	4	+4					
LSR	Logical Shift Right	4A	--	46	4E	56	5E	--	--	--	--	N.....ZC
	(przesuń w prawo)	2		5	6	6	7					
ORA	Or Accumulator	--	09	05	0D	15	1D	--	19	01	11	N.....Z.
	(logiczne "lub")		2	3	4	4	+4		+4	6	+5	
ROL	Roll Byte Left	2A	--	26	2E	36	3E	--	--	--	--	N.....ZC
	(obróć w lewo przez C)	2		5	6	6	7					
ROR	Roll Byte Right	6A	--	66	6E	76	7E	--	--	--	--	N.....ZC
	(obróć w prawo przez C)	2		5	6	6	7					
SBC	Subtract with Borrow	--	E9	E5	ED	F5	FD	--	F9	E1	F1	NV....ZC
	(odejmij z pożyczką)		2	3	4	4	+4		+4	6	+5	
STA	Store A register	--	--	85	8D	95	9D	--	99	81	91	....
	(zapamiętaj akumulator)			3	4	4	5		5	6	6	
STX	Store X register	--	--	86	8E	--	--	96	--	--	--	....
	(zapamiętaj rejestr X)			3	4			4				
STY	Store Y register	--	--	84	8C	94	--	--	--	--	--	....
	(zapamiętaj rejestr Y)			3	4	4						

OBJAŚNIENIA:

1. + oznacza konieczność dodania jednego cyklu zegarowego, gdy dodanie indeksu powoduje zmianę strony pamięci.

2. Działanie instrukcji BIT:

Z ← (A)^(M)

N ← (M)bit 7

V ← (M)bit 6

3. Działanie obrotów i przesunięć:

ASL C ← | 7.........0 | ← 0

LSR 0 → | 7.........0 | → C

ROL C ← | 7.........0 | ← C

ROR C → | 7.........0 | → C

nie 8-bitowego adresu (są dwubajtowe) i są wykonywane szybciej. Przy projektowaniu typowego oprogramowania stronę zerową wykorzystuje się do przechowywania najczęściej używanych danych.

Również na zasadzie skróconego adresowania jest oparte działanie wspomnianego wcześniej 256-bajtowego stosu. Dla niego jest zarezerwowana strona nr 1 pamięci, czyli adresy od 100_H do $1FF_H$.

7							0
N	V	-	B	D	I	Z	C

Sześć ostatnich lokacji pamięci zawiera adresy procedur obsługi przerwań, w następującej kolejności:

$FFFA_H$, $FFFB_H$ – NMI
$FFFC_H$, $FFFD_H$ – Reset
$FFFE_H$, $FFFF_H$ – IRQ i instrukcja BRK

Po przyjęciu przerwania 6502 automatycznie przesyła na stos zawartości licznika rozkazów i rejestru wskaźników. Podczas wykonania instrukcji powrotu z przerwania RTI są odtwarzane zapamiętane wartości. Instrukcja BRK pełni rolę przerwania programowego, wykorzystywanego przede wszystkim podczas uruchamiania oprogramowania. Odróżnienie, czy przerwanie zostało wywołane z zewnątrz, poprzez linię IRQ, czy też nastąpiło na skutek wykonania instrukcji BRK, wymaga testowania stanu wskaźnika B.

Tablice zawierają zestawienie trybów adresowania i kompletną listę instrukcji 6502.

Uważna analiza obu tablic prowadzi do kilku istotnych wniosków. Przede wszystkim – 6502 oferuje stosunkowo niewiele rozkazów, ale w połączeniu z wieloma trybami adresowania są one bardzo efektywne. Przyjęte nazwy rozkazów są logiczne – bezbłędnie kojarzą się z wykonywanymi operacjami. Ułatwia to nauczenie się rozkazów i swobodne ich używanie bez konieczności sięgania do ściągawki. Ponadto lista rozkazów 6502 wykazuje małą nadmiarowość – oznacza to, że rzadko kiedy typowy prosty problem można rozwiązać na kilka sposobów. Zazwyczaj dla typowego zadania programowego istnieje jedno dobre rozwiązanie, co pozwala nie zastanawiać się za każdym razem nad możliwościami zakodowania elementarnego algorytmu krócej i szybciej.

Dobremu programiście 6502 sam narzuca dobry styl programowania. Wiąże się to z faktem, że jego architektura została dostosowana do listy rozkazów, zaprojektowanej przez programistów współuczestniczących w pracach konstrukcyjnych. W przypadku innych, popularnych u nas mikroprocesorów wszystko wskazuje na to, że najpierw zaprojektowano architekturę wewnętrzną, a potem tak konstruowano listę rozkazów, aby jak najlepiej ją wykorzystać – zgodnie z zasadą "co by tu jeszcze dołożyć".

Reasumując: 6502 jest łatwo programować efektywnie i w dobrym stylu. Na poparcie twierdzenia o jego zaletach warto przytoczyć zaskakujący na pierwszy rzut oka fakt, często wspominany w literaturze. Otóż popularne mikrokomputery, oparte na 6502 pracującym z zegarem o częstotliwości 1 MHz, najczęściej nie są wolniejsze od tej samej klasy komputerów zbudowanych na Z80 z zegarem 4 MHz. A 6502 jest też produkowany w wersji 3 MHz.

W następnej, już ostatniej części cyklu o 6502 zostaną przedstawione typowe techniki programowania.

MACIEJ KASPERSKI

LISTA ROZKAZÓW 6502

1. Skoki względne

BPL	Branch if Plus	(skocz, gdy N=0)	10	2 cykle, gdy warunek nie spełn.
BMI	Branch if Minus	(skocz, gdy N=1)	30	
BVC	Branch if V Clear	(skocz, gdy V=0)	50	
BVS	Branch if V Set	(skocz, gdy V=1)	70	3 cykle, gdy skok na tej samej str.
BCC	Branch if C Clear	(skocz, gdy C=0)	90	
BCS	Branch if C Set	(skocz, gdy C=1)	B0	4 cykle, gdy skok do innej strony
BNE	Branch if Not Eq.	(skocz, gdy Z=0)	D0	
BEQ	Branch if Equal	(skocz, gdy Z=1)	F0	

2. Skoki bezwzględne

JMP	Jump unconditional	(skok bezwarunkowy)	4C	3
JMP	unconditional Jump indirect	(... pośredni)	6C	5
JSR	Jump to Subroutine	(wywołanie procedury)	20	6

3. Instrukcje w trybie implikowanym

BRK	Break	(przerwanie programowe)	00	7	. . . B . I . .
CLC	Clear Carry	(zeruj przeniesienie)	18	2	 C
CLD	Clear Dec. Mode	(ustaw tryb binarny)	D8	2	. . . . D . . .
CLI	Clear I Mask	(zezwól na przerwania)	58	2	. . . . . I . .
CLV	Clear Overflow	(zeruj znacznik V)	B8	2	. V . . . . . .
DEX	Decrement X-reg.	(zmniejsz zaw. X)	CA	2	N . . . . . Z .
DEY	Decrement Y-reg.	(zmniejsz zaw. Y)	88	2	N . . . . . Z .
INX	Increment X-reg.	(zwiększ zawartość X)	E8	2	N . . . . . Z .
INY	Increment Y-reg.	(zwiększ zawartość Y)	C8	2	N . . . . . Z .
NOP	No Operation	(nic nie rób)	EA	2	. . . .
PHA	Push A-register	(zawartość A na stos)	48	3	. . . .
PHP	Push Program Stat.	(wskaźniki na stos)	08	3	. . . .
PLA	Pull A from Stack	(odtwórz A ze stosu)	68	4	N . . . . . Z .
PLP	Pull Pgm status	(odtwórz wskaźniki)	28	4	odtworzone
RTI	Return from Intr.	(wróć z przerwania)	40	6	odtworzone
RTS	Return from Sbr.	(wróć z podprogramu)	60	6	. . . .
SEC	Set Carry	(ustaw wskaźnik C)	38	2	 C
SED	Set Decimal Mode	(ustaw tryb dziesiętny)	F8	2	. . . . D . . .
SEI	Set I Mask	(zablokuj przerwania)	78	2	. . . . . I . .
TAX	Transfer A to X	(prześlij A do X)	AA	2	N . . . . . Z .
TAY	Transfer A to Y	(prześlij A do Y)	A8	2	N . . . . . Z .
TSX	Transfer S to X	(prześlij S do X)	BA	2	N . . . . . Z .
TXA	Transfer X to A	(prześlij X do A)	8A	2	N . . . . . Z .
TXS	Transfer X to S	(prześlij X do S)	9A	2	. . . .
TYA	Transfer Y to A	(prześlij Y do A)	98	2	N . . . . . Z .

Uwaga: wszystkie rozkazy są podawane według schematu –
KOD – znaczenie – (tłumaczenie) – kod hex – ilość cykli zeg. – ewent. aktywne wskaźniki

TRYBY ADRESOWANIA 6502

nazwa trybu angielska	polska	sposób zapisu argumentu	wartość efektywnego adresu argumentu
Immediate	natychmiastowe	n	PC+1
absolute	pełne	nn	(PC+2):(PC+1)
zero page	strony zerowej	n	0:(PC+1)
indirect absolute	pośrednie pełne	(nn)	((PC+2):(PC+1))
absolute indexed X	pełne indeksowane X	nn,X	(PC+2):(PC+1)+X
absolute indexed Y	pełne indeksowane Y	nn,Y	(PC+2):(PC+1)+Y
zero page indexed X	strony zerowej indeksowane X	n,X	0:(PC+1)+X
zero page indexed Y	strony zerowej indeksowane Y	n,Y	0:(PC+1)+Y
indexed indirect	indeksowane pośrednie	(n,X)	(0:(PC+1)+X)
indirect indexed	pośrednie indeksowane	(n),Y	(0:(PC+1))+Y
relative	względne	±n	PC+(PC+1)
accumulator	akumulatora	A	-
implied	implikowane	-	-

OZNACZENIA: n – wartość 8-bitowa
nn – wartość 16-bitowa
PC, X, Y – zawartości odpowiednich rejestrów

Prawa Murphy'ego

"Im program dłuższy, tym mniej sprawny"

Jeśli jesteś absolwentem informatyki lub właśnie opracowujesz system operacyjny dla skonstruowanego przez siebie 32-bitowego mikrokomputera, zaniechaj czytania tego tekstu. Nic nowego w nim nie znajdziesz. Adresowany jest on przede wszystkim do tych, którzy spędzając pierwsze noce przy komputerze piją mocną kawę, czerwonymi ze zmęczenia oczyma wpatrują się w ekran i pytają, dlaczego ten piekielny program wciąż nie chce chodzić, mimo znalezienia w nim już i usunięcia kilku grubych błędów.

Ostatecznie w porównaniu z innymi językami BASIC jest śmiesznie prosty. Brak w nim jakichkolwiek nieregularnych czasowników, brak skomplikowanej gramatyki – trzeba tylko znać kilkadziesiąt prostych rozkazów, na które komputer reaguje zawsze szybko i pewnie.

Ufając powodzeniu swoich pierwszych krótkich programików stawiasz sobie teraz Czytelniku trochę bardziej ambitne zadanie. Ideę programu poniekąd masz już w głowie, wszystkie instrukcje są wprowadzone prawidłowo, a jednak Twój program wcale nie robi tego, co powinien.

Chciałbym Cię jednak pocieszyć; obok pana Murphy'ego, autora takich fundamentalnych praw jak to, że "Kromka chleba posmarowana marmoladą zawsze spadnie na podłogę stroną posmarowaną", fenomenem tym zajmowało się tysiące innych sfrustrowanych programistów: dlaczego programy mają zawsze więcej błędów, aniżeli można by się tego spodziewać?

Chcąc wpaść na trop przyczyn, trzeba wiedzieć, że różnica między długim a krótkim programem nie ma charakteru ilościowego, lecz raczej jakościowy. Długość programu bynajmniej nie rośnie "liniowo", lecz nieuchronnie wraz z każdą nową linią programu liczba rozwidleń (IF..THEN, GOTO) i zmiennych pomnaża się. Końcowy efekt jest podobny jak przy gotowaniu spagetti: początkowy porządek po zamieszaniu łyżką przeistacza się w nieprzejrzany splot nitek makaronu.

Sławny holenderski informatyk profesor E.W. Dijkstra próbował zilustrować ten fenomen formułą P=p^ N: Prawdopodobieństwo, że program jest bezbłędny (P), zależy od prawdopodobieństwa błędu (p) podniesionego do potęgi całkowitej liczby komponentów (N). O czym myślał profesor Dijkstra, pokazuje przykład. Weźmy programującego "geniusza", który na sto decyzji popełnia tylko jeden błąd. Prawdopodobieństwo popełnienia przezeń błędu jest więc równe 0.99. Jeżeli teraz program składa się z dziesięciu komponentów, to prawdopodobieństwo, że będzie działał od pierwszego "run", wynosi 0.99 ^ 10, czyli 09. Jeśli nie jesteś geniuszem i przy 100 decyzjach popełniasz 10 błędów, to wartość ta spada do 0.9 ^ 10=0.35. Przy stu komponentach (0.9 ^ 100) prawdopodobieństwo uzyskania od razu bezbłędnego programu wynosi tylko 0.000027. Możesz się więc przy pomocy formuły profesora Dijkstry pocieszyć: szansa, że Twój program zacznie działać bezbłędnie od pierwszego razu jest mniej więcej taka, jak możliwość bycia trafionym w czasie burzy raz po razie piorunem.

Formuła Dijkstry mówi Ci także, jak pisać lepsze programy: należy "tylko" zmniejszyć prawdopodobieństwo błędu. Jak do tego dojść, przeczytać można w wielu fachowych książkach. Rozwiązywany problem powinien zostać najpierw poddany gruntownej analizie. Potem należy wykonać wyczerpujący schemat działania programu. Dopiero po takim przygotowaniu można rozpocząć kodowanie tzn. wprowadzanie tekstu programu do komputera. Zakładając oczywiście, że znasz maszynę i jej język programowania.

Od paru tygodni jesteś więc dumnym posiadaczem komputera domowego i poznałeś właśnie najważniejsze funkcje i rozkazy urządzenia i języka programowania. Znajdujesz się w sytuacji pilota, który pali się do latania dopiero co poznanym samolotem nie wiedząc, dokąd ma lecieć!

W tej sytuacji mogą Ci być pomocne pewne, wynikające z doświadczeń pokoleń programistów, reguły programowania.

REGUŁA 1.: Przy numerowaniu kolejnych instrukcji (linii programu) ustal krok wzrastania numerów z pewnym zapasem, tzn. nie co jeden, lecz np. co dziesięć. Poza tym zrób w numeracji coś na wzór "szuflad". Każda "szuflada" może obejmować 50, 100, 200 lub nawet 1000 numerów linii. Dla głównego modułu sterującego zarezerwuj 1000 linii, czyli np. od 4000 do 4999. Każdy podprogram i każda część Twojego programu głównego zaczynać się będzie dzięki temu od pełnej setki i odpowiedniego komentarza.

REGUŁA 2.: Skreśl ze słownika swojego języka BASIC instrukcję GOTO! GOTO jest bowiem jak czarodziejski latający dywan: przy pomocy tego rozkazu możesz skoczyć do dowolnego punktu swojego programu. Fascynująca możliwość. Niestety, dzięki temu każdy program przekształca się prędko w nieprzenikniona dżunglę. Przed tym uchroni Cię instrukcja GOSUB (lub ON n GOSUB x, y, z), której RETURN spowoduje zawsze powrót do punktu wyjścia. Jest to szczególnie pomocne przy poszukiwaniu błędów.

REGUŁA 3.: Te części programu, które są często używane, powinieneś umieścić w podprogramach. Przy programach dłuższych dobrze jest korzystać z podprogramów, gdyż przy tropieniu błędów ułatwia to poznanie wartości przejściowych.

REGUŁA 4.: Zanotuj sobie, co oznaczają nazwy zmiennych, których używasz. W przeciwieństwie do innych języków programowania, zmienne w BASICU są globalne, tzn. dostępne dla całego programu, a nie tylko dla konkretnego podprogramu. Powinieneś zatem np. nazwy dx używać tylko dla określenia przyrostu zmiennej x lub współrzędnej odciętej punktu. Jest to trywialne, logiczne i dlatego tym bardziej denerwujące, gdy trzeba później sporo czasu poświęcić, aby odnaleźć takie złośliwe błędy. Jeżeli dialekt Twojego języka BASIC dopuszcza dłuższe nazwy zmiennych, to zrób z tego użytek. Poświęcisz więcej czasu na wprowadzenie programu do pamięci komputera, ale będziesz miał większą pewność, że nie użyłeś dwa razy tej samej nazwy zmiennej.

REGUŁA 5.: Pamiętaj o tym, że programy nie są zapamiętywane przez Twoją pamięć (tę we własnej głowie) w taki sam sposób jak na dyskietce. Program, z którego podczas pisania pamiętasz każdą instrukcję i każdą zmienną nawet podczas snu, może Ci się już po kilku tygodniach wydać nieznanym labiryntem. Obok przejrzystego podziału programu, jednoznacznych i dobrze udokumentowanych nazw zmiennych szybko odnaleźć się w programie pomoże Ci na pewno wydruk, wykonany i opisany natychmiast po zakończeniu pracy.

Być może uda Ci się przy pomocy powyższych i pięćdziesięciu innych reguł, które podsunie Ci Twoja praktyka, zaprzeczyć profesorowi Murphy'emu: Twoje programy choć długie, będą robić to, co powinny.

Przecież jest jeszcze reguła Dijkstry!

Opr. TOMASZ ZIELIŃSKI



Pierwsze kroki z IBM PC

W 1980 roku IBM rozpoczął produkcję 16-bitowego personala – IBM PC, mikrokomputera przeznaczonego głównie dla biur. PC zbudowany był na zupełnie innym procesorze niż zadomowiony na rynku mikrokomputerowym Apple, wymagało to napisania od początku całego oprogramowania. IBM, jako jedno z największych przedsiębiorstw amerykańskich o wielkich doświadczeniach software'owych, mógł podołać tak olbrzymiej pracy dystansując wszystkich łącznie z Applem.

Do wielkiego sukcesu przyczyniło się kilka dobrych posunięć menegerów IBM. Uznali, że komputer aby znalazł wielu nabywców musi mieć wielką różnorodność oprogramowania, bez tego nie wejdzie się na rynek opanowany przez innych. Opublikowano więc dokładny opis systemu operacyjnego DOS z opisem funkcji, co pozwoliło innym programistom nie związanym z producentem na tworzenie systemów omijających funkcje DOS lub korzystających z nich. Spowodowało to dodatkowy (a może głównie o to chodziło) efekt: mniejsze firmy zaczęły kopiować – czasami ulepszając – IBM PC zapewniając swoim produktom bogate oprogramowanie zgodne z IBM PC. Został stworzony pewien standard (przyjmują go też kraje RWPG) – standard IBM. W krajach socjalistycznych rozpoczęto produkcję mikrokomputerów kompatybilnych z wzorcem standardu (w ZSRR-Iskra, w Bułgarii – Pravec, Węgry – Proper 16). Również w naszym kraju niedługo zostanie rozpoczęta produkcja odpowiedników IBM PC, wrocławska ELWRO 800 i warszawska – Mazovia 1016. Tymczasem wiele firm polonijnych składa personale z tajwańskich części. Ilość PC (w tym tajwańskich kopii) w naszym kraju szybko wzrasta. W wielu biurach już są, a w wielu będą po rozpoczęciu produkcji polskiego PC. Pojawia się problem nauczenia obsługi nieskomplikowanego, ale jednak zupełnie nowego sprzętu, jak również posługiwanie się oprogramowaniem.

Wszyscy ci, którzy po raz pierwszy zetknęli się z profesjonalnym mikrokomputerem, ze zdziwieniem patrzyli na pusty ekran monitora i kilka grubych ksiąg opisujących działanie całego systemu. Wielu w tym momencie zrezygnowało z posługiwania się komputerem, inni zaś poważnie się zniechęcili. Trzysta stron opisu systemu operacyjnego, kilka opasłych tomów dotyczących programów użytkowych może wielu odstraszyć. Jeśli skończyłeś naukę kilka lat temu, nie zawsze będziesz miał ochotę zostać uczniakiem wkuwającym z podręcznika rozdział po rozdziale. A jaka jest alternatywa? Nie korzystać z komputera? Czy programiści tworzący systemy operacyjne, programy użytkowe i opisy do nich każą nam wertować tysiące stron zanim będziemy mogli dotknąć klawiatury? Oczywiście, że nie. Każda szanująca się firma dba o to, aby jej program można było poznać i korzystać z niego w kilka dni. Do większości pakietów oprogramowania dołączany jest też dodatkowo program – nauczyciel.

Firma IBM dołącza do sprzedawanego komputera, oprócz standardowego zestawu oprogramowania, program nauczający. Omawia on pokrótce podstawową obsługę, opisuje sprzęt i oprogramowanie. Nie ma na celu nauczania, a raczej przygotowanie do podjęcia nauki z podręcznika, uczy selektywnego podejścia do przyswajanej wiedzy. Nie każdy jednak lubi taką formę uczenia się, jest ona czasochłonna i niezbyt pewna.

Dla tych, którzy przedkładają naukę z komputerem o komputerze nad czytanie książek, proponuję pakiet angielskiej firmy S. R. A. "Computer Based Trainig" składający się z pięciu dyskietek: "Introduction to the IBM Personal Computer", "Hardware for IBM PC/XT", "Introduction to the BASIC Programing Language", "Using the IBM Personal Computer Hardware", "The IBM Personal Computer Disk Opereting System". Pakiet ten jest podstawowym kursem dla początkujących.

Każda dyskietka to jakby osobna książka. Wszystkie mają strukturę zwykłego podręcznika, a więc uwzględniony jest podział na rozdziały, rozdziały zaś dzielą się na ćwiczenia, a te z kolei zawierają zazwyczaj kilka pytań do omówionego materiału. Zarówno korzystając z podręcznika jak i w przypadku posługiwania się komputerem można wybiórczo podejść do przyswajanych wiadomości. Jakie są wobec tego różnice? Pierwsza i zasadnicza to fakt, że osoba zainteresowana siedzi przed komputerem, który zaznajamia ją z własną strukturą. Drugą stanowi to, że w procesie uczenia się – nauczania, komputer spełnia funkcję podawczo-informacyjną (jak podręcznik), a także objaśniająco-nadzorującą (jak nauczyciel).

Pierwsze co należy obejrzeć (i przećwiczyć) to zawartość dyskietki "Introduction to the IBM Personal Computer". Pierwszy rozdział to łatwa w posługiwaniu się instrukcja do kursu. Drugi zawiera podstawowe wiadomości o komputerze – trochę historii, trochę o elektronicznym przetwarzaniu danych, o językach programowania i systemach operacyjnych oraz o tym jak współpracują ze sobą sprzęt i oprogramowanie. Ostatni, to krótki przewodnik po systemach operacyjnych oraz językach programowania (a właściwie kompilatorach i interpreterach).

Pozostałe dyskietki zawierają rozwinięcie tematów zasygnalizowanych w drugim i trzecim rozdziale. Na każdą dyskietkę trzeba poświęcić około 2,5 godz. Te które naprawdę trzeba przeczytać, to wstępna, o której krótko wspomniałem, "The IBM Personal Computer Disk Opereting System" i trzecia "Using the IBM Personal Computer Hardware". Po obejrzeniu ich możesz rozpocząć poznawanie programów użytkowych.

LECH BĄK

Akademia KOMPUTERA

Wykład inauguracyjny: „Jak uzyskać pełną zgodność i zrozumienie między komputerami różnych firm naśladujących IBM PC (i nie tylko), czyli jak rozmawiać o kompatybilności".

CP/M, to system operacyjny zupełnie niezgodny z PC-DOS.

CP/M 86 – system operacyjny niezgodny zupełnie nie tylko z PC-DOS, ale i CP/M.

Kompatybilny – po polsku znaczy zgodny.

MS-DOS – niby to samo co PC-DOS, ale ... produkt tej samej firmy Microsoft nie musi być tym samym, co okazuje się zazwyczaj w momencie najbardziej irytującym.

PC-DOS – system operacyjny dla IBM PC i pochodnych.

Najnowszy – koncepcja, która nie przybrała jeszcze kształtu prototypu.

Nowy – praktycznie nie do zdobycia dla zwykłego klienta.

Potężny – całkiem niezrozumiały i z niczym niezgodny.

Rewolucyjny – niezgodny z czymkolwiek.

RS-232 – standard przemysłowy na łącze szeregowe między komputerami. Patrz: standard przemysłowy.

Standard przemysłowy: patrz "zgodny".

Szeroko stosowany – zdecydowanie przestarzały.

Zgodny – kompatybilny z nazwy, czasem też z działania.

Zgodny w pełni – z początku nie działa, lecz na ogół wystarczy Ci kilka minut, by odkryć, na czym polega nadmierny optymizm lub oszustwo producenta.

Wtyczko-zgodny (plug-compatible) – jest szansa, że po uruchomieniu okaże się rzeczywiście zgodny, choć zapewne Twoje kłopoty nie skończą się na prawidłowym połączeniu sznurów i wtyczek.

Wniosek: Nie kupuj kota w worku! Jedyną szansą zdobycia pewności, że coś z czymś będzie zgodnie współpracowało, jest przekonać się o tym osobiście i zobaczyć, jak to rzeczywiście pracuje. Prawa reklamy są inne niż prawa życia, a w naszym kraju odpowiedzialność za reklamę bez pokrycia praktycznie nie istnieje.

Jeśli "kot" ma być zgodny z IBM PC, dobrym testem jest sprawdzenie, czy bez zacięć pracuje na nim program "Flight Simulation" firmy Microsoft.

(Rady The Observer spolszczył W. Majewski)

Zgadywanka

Czym zadziwić znajomych, którzy nas odwiedzili? Jak zaprezentować swój dom? Dawniej to było łatwe: wystarczyło sprowadzić Cyganów z oswojonym misiem czy marynarza z tresowaną małpką. Teraz takie rozrywki nie są w modzie. Każdy widział w ZOO nie tylko małpkę, ale nawet krokodyla. Trzeba wymyśleć coś odpowiedniego do naszego XX wieku – już nie pary i elektryczności, ale atomu i... – no właśnie – komputerów!

Jeżeli mamy pod ręką jakiś komputer, ot zwykłe Spectrum czy Commodore, to możemy zaprezentować „inteligentną maszynę" mądrzejszą od niejednego z nas. To nie dowcip! Któż z nas potrafi w sześciu pytaniach zgadnąć np. państwo europejskie, które sobie pomyślałem, jeżeli na zadawane mi pytania odpowiadam tylko „tak" lub „nie"? A komputer potrafi. Dzieje się to tak: uruchamiamy komputer i wprowadzamy program. Maszyna proponuje „Pomyślcie sobie jakieś państwo europejskie". Porozumiewając się na migi, aby komputer nie mógł podsłuchać, decydujemy się np. na Francję. Teraz komputer zadaje pytania. Po kilku zaczyna być widoczne, że maszyna już coś podejrzewa, że już chyba wie, a tylko sprawdza naszą wiedzę. W końcu ostatnie pytanie: „Czy tym państwem jest Francja?" Dobra – mówimy sobie – to było łatwe. Spróbujmy Islandię – zgadł, San Marino – też zgadł. W dodatku za każdym razem pytał o coś innego, więc chyba rzeczywiście myśli.

Wszystkie okrzyki zachwytu i wiwaty pod adresem komputera przyjmie na siebie z lekkim uśmieszkiem... programista. Mówcie sobie co chcecie, ale to przecież ja pożyczyłem nieco własnej inteligencji tej plątaninie drutów. To ja zastosowałem algorytm binarnego wyszukiwania informacji, a zestaw pytań dobrałem tak, aby każda odpowiedź „tak" lub „nie" wskazywała tę z dwóch grup państw, w której znajduje się pomyślany przez zgadujących kraj.

Najprościej ten algorytm można zrealizować tak: stosujemy jedno tylko pytanie „Czy pomyślany kraj jest między wymienionymi: ..." i tu wypisujemy połowę nazw wszystkich państw w Europie. Po odpowiedzi „tak", z tym samym pytaniem wypisujemy już tylko połowę uprzednio wypisanych i tak dalej. Jeżeli pada odpowiedź „nie", to zapominamy o tych wszystkich państwach, które były już podane i wypisujemy połowę tych, które zostały, i tak dalej. Kończymy to postępowanie, gdy odpowiadający sam wskaże drogę dojścia do tego jedynego państwa.

Niestety, już przy trzecim pytaniu nasi goście zaczynają się nudzić, a co bardziej nerwowi wietrzą jakiś podstęp czy oszustwo. Inni proponują, by obejrzeć „idących" właśnie w TV Fragglesów. Aby temu zapobiec – dołóżmy nieco starań, żeby pytania nie były monotonne. Niech nazwa zgadywanego państwa pojawi się dopiero wtedy, gdy komputer pochwali się, że już zgadł. Wywołamy efekt, że jednak pomyślał.

Przystępujemy do układania pytań. Zauważmy, że z 34 państw w Europie, dokładnie 17 (a więc połowa) ma terytorium mniejsze niż 100 tys. km^2. Czyli pierwsze pytanie powinno brzmieć: „Czy powierzchnia tego państwa jest mniejsza niż 100 tys. km^2?" Dalej łatwo zauważyć, posługując się mapą, że 9 spośród małych państw nie ma dostępu do morza, a więc... pytamy o to. Teraz wykorzystajmy encyklopedię, z której na przykład można wyczytać, że 4 wśród małych państw z dostępem do morza, to monarchie, a dwie z tych monarchii mają trójkolorowe flagi, itd.

Zestaw pytań proponuję w zamieszczonym obok programie, w którym dla oszczędności tak je dobrałem, by zamiast 64 pytań wystarczyło 15 odpowiedzi modyfikowanych.

Przedstawiony tu program może służyć do zgadywanki z dowolnej dziedziny. Trzeba tylko zmienić linie danych. Pamiętajmy przy tym, że odpowiedź na każde pytanie musi wyznaczać grupę zgadywanych obiektów zawierającą połowę obiektów z grupy wyznaczonej uprzednio. W takiej sytuacji łatwo policzyć, że jeżeli mamy n obiektów, to stosując binarne wyszukiwanie wystarczy zadać k pytań, gdzie $2^{k-1} < n < 2^k$.

LESZEK RUDAK

```
   1 REM   PROGRAM ZGADYWANKA
   2 REM     LESZEK RUDAK
   9 REM W TABLICY T PRZECHOWUJE
MY NUMERY PYTAN
  10 DIM T(63): FOR I=1 TO 63: R
EAD T(I): NEXT I
  14 REM LINIA 15  TYLKO DLA ZX
SPECTRUM, DLA MERITUM NALEZY JA
USUNAC
  15 POKE 23658,8
  20 CLS
  30 PRINT "POMYSLCIE SOBIE JAKI
ES PANSTWO  EUROPEJSKIE..."
  35 PRINT " ODPOWIADACIE TAK lu
b NIE"
  40 INPUT "JESTESCIE GOTOWI ?";
A$
  49 REM N WYZNACZA MIEJSCE SZUK
ANEGO PANSTWA NA NASZEJ LISCIE
  50 LET N=0
  59 REM ZADAJEMY 6 PYTAN
  60 FOR I=0 TO 5
  69 REM M BEDZIE PARMETREM PODP
ROGRAMU
  70 LET M=T(2↑I+N)
  79 REM SZUKAMY PYTANIA
  80 GO SUB 300
  90 PRINT A$
 100 INPUT B$
 109 REM KONTROLA POPRAWNOSCI OD
POWIEDZI
 110 IF B$<>"TAK" AND B$<>"NIE"
THEN  GO TO 90
 119 REM TU ISTOTA BINARNEGO WYS
ZUKIWANIA
 120 LET N=2*N
 130 IF B$="TAK" THEN  LET N=N+1
 140 NEXT I
 149 REM PRZYGOTOWANIE PARAMETRU
...
 150 LET M=N+17
 160 GO SUB 300
 170 PRINT "  POMYSLANYM PANSTWE
M JEST              ";A$
 180 INPUT "GRAMY JESZCZE RAZ ?"
;B$
 190 IF B$="TAK" THEN  GO TO 20
 200 STOP
 299 REM PODPROGRAM WYSZUKUJE M-
TA DANA I PODSTAWIA JA NA A$
 300 RESTORE 540
 301 REM LINIE 300 I 310 SA DLA
  SPECTRUM. DLA MERITUM POWINNY
  MIEC POSTAC :
   300 RESTORE
   310 FOR J=0 TO M+62
 310 FOR J=0 TO M-1
 320 READ A$
 330 NEXT J
 340 RETURN
 499 REM LICZBY DO TABLICY T
 500 DATA 5,3,8,2,7,16,4,4,12,10
,11,4,9,6
 510 DATA 14,9,9,7,15,8,10,13,3,
4,14,12,3,6,7,15
 520 DATA 1,1,1,1,13,1,1,13,1,1,
1,1,1,1,1,1
 530 DATA 1,1,2,1,1,1,1,1,1,1,1,
1,1,1,1,1
 539 REM TEKSTY PYTAN
 540 DATA "CZY SA TAM CZARNE KOT
Y ?"
 550 DATA "CZY STOLICA MA MNIEJ
NIZ MILION MIESZKANCOW ?"
 560 DATA "CZY JEST TO PANSTWO S
OCJALISTYCZNE ?"
 570 DATA "CZY JEST TO MONARCHIA
 ?"
 580 DATA "CZY POWIERZCHNIA TEGO
 PANSTWA JEST MNIEJSZA NIZ 10000
0 KM KWADRATOWYCH ?"
 590 DATA "CZY MA TROJKOLOROWA F
LAGE ?"
 600 DATA "CZY MA DOSTEP DO BALT
YKU ?"
 610 DATA "CZY MA DOSTEP DO MORZ
A ?"
 620 DATA "CZY STOLICA LEZY NA W
YSPIE ?"
 630 DATA "CZY JEST TO PANSTWO W
IELONARODOWOSCIOWE ?"
 640 DATA "CZY CALE LEZY W INNYM
 PANSTWIE ?"
 650 DATA "CZY JEST TO PANSTWO S
KANDYNAWSKIE ?"
 660 DATA "CZY PIENIADZ W TYM KR
AJU TO MARKA ?"
 670 DATA "CZY GRANICZY Z AUSTRI
A ?"
 680 DATA "CZY KWIATY MAJA DUZE
ZNACZENIE W GOSPODARCE TEGO KRAJ
U ?"
 690 DATA "CZY GRANICZY Z FRANCJ
A ?"
 700 DATA "FRANCJA","FRANCJA","W
LOCHY","WLOCHY","HISZPANIA","HIS
ZPANIA","ANGLIA","ANGLIA"
 710 DATA "GRECJA","RFN","ISLAND
IA","ISLANDIA","NORWEGIA","NORWE
GIA","SZWECJA","FINLANDIA"
 720 DATA "RUMUNIA","RUMUNIA","B
ULGARIA","BULGARIA","CZECHOSLOWA
CJA","CZECHOSLOWACJA","JUGOSLAWI
A","JUGOSLAWIA"
 730 DATA "POLSKA","POLSKA","ZSR
R","ZSRR","NRD","NRD","NRD","NRD
"
 740 DATA "AUSTRIA","LIECHTENSTE
IN","WEGRY","WEGRY","BERLIN ZACH
ODNI","SAN MARINO","WATYKAN","WA
TYKAN"
 750 DATA "ANDORA","ANDORA","SZW
AJCARIA","SZWAJCARIA","LUKSEMBUR
G","LUKSEMBURG","LUKSEMBURG","LU
KSEMBURG"
 760 DATA "PORTUGALIA","PORTUGAL
IA","ALBANIA","ALBANIA","MALTA",
"MALTA","IRLANDIA","IRLANDIA"
 770 DATA "MONAKO","MONAKO","DAN
IA","DANIA","BELGIA","BELGIA","H
OLANDIA","HOLANDIA"
```

Zakładamy Klub Mistrzów KOMPUTERA – otwarty zarówno dla Czytelników mających własny sprzęt lub dostęp do niego, jak i dla mogących o nim tylko marzyć. Wystarczy nie bać się myślenia.

Nasze hasło to "kości mózgu nie zastąpią!", skostniałe głowy, nie potrafiące ruszać się samodzielnie – to nie my!

Do klubu zapraszamy wszystkich, którzy:

1. rozwiążą i wyślą w terminie 3 miesięcy 6 spośród 9 publikowanych w kolejnych numerach „Komputera" specjalnie oznaczonych zadań klubowych (trzech w każdym numerze)

2. ułożą i przyślą do redakcji co najmniej dwa zadania, które zostaną włączone do banku zadań Klubu.

Pierwszą grupę członków Klubu powoła redaktor naczelny jesienią br., a ich lista zostanie opublikowana w numerze grudniowym. Uwzględniać będziemy przy rozpatrywaniu kandydatur także pozakonkursowe osiągnięcia w zakresie poznawania, stosowania i propagowania informatyki. Liczymy, że klubowicze staną się współpracownikami redakcji, dzielącymi się z nami swymi uwagami, propozycjami i spostrzeżeniami dotyczącymi sprzętu i oprogramowania.

Członkostwo Klubu nie będzie, niestety, dożywotnie: aby je zachować, spełniać trzeba będzie co roku co najmniej połowę wymagań potrzebnych do jego zdobycia.

Członkowie Klubu mają prawo do:

– bezpłatnej prenumeraty „Komputera" i broszur z serii ABC „Komputera"

– korzystania z zasobów sprzętowych i książkowych redakcji

– uczestnictwa w dorocznym turnieju o tytuł informistrza

– noszenia odznaki klubowej

Redakcja będzie się starać o uzyskanie dla członków Klubu dalszych przywilejów.

Korespondencję w sprawach Klubu prosimy oznaczać dopiskiem na kopercie KLUB KOMPUTERA. Inicjatorem Klubu i redaktorem zadań jest – z upoważnienia redakcji – Leszek Rudak.

KARUZELA

Obrazki przypominają dobrze znane efekty działania programu VU-3D, stworzył je jednak programik w skondensowanej wersji mieszczącej się w 40 wierszach języka BASIC. Jego najbardziej szokującej możliwości nie sposób jednak przedstawić w statycznej ilustracji: bryły te są... ruchome. Obracają się wokół swej pionowej osi w tempie ok. 40 obrotów na minutę! Powstaje złudzenie ruchu nie gorsze niż przy oglądaniu TV. Jest to oczywiście możliwe tylko dzięki ich wysokiej symetrii, gdyż cały ruch jest wynikiem złożenia tylko sześciu ekranów, jakie mieści pamięć ZX Spectrum.

W pierwszej linii po słowie REM należy wpisać 12 dowolnych znaków – w to miejsce program ładujący zawarty w liniach 15-50 wpisze 12 bajtów kodu maszynowego służącego do szybkiego przerzucania pamięci ekranu (instrukcja LDIR). Po uruchomieniu programu i wpisaniu kodu do pierwszej linii – linie 15-50 można usunąć.

Następnie projektujemy przekrój poprzeczny naszej bryły. Program sam buduje sześć jego rzutów widzianych pod różnymi kątami, ładuje gotowe obrazki do pamięci, po czym (linia 440) zmienia kierunek przesyłania obrazów – i możemy (linia 450) puścić nasze małe kino w ruch. Wrażenie fascynujące!

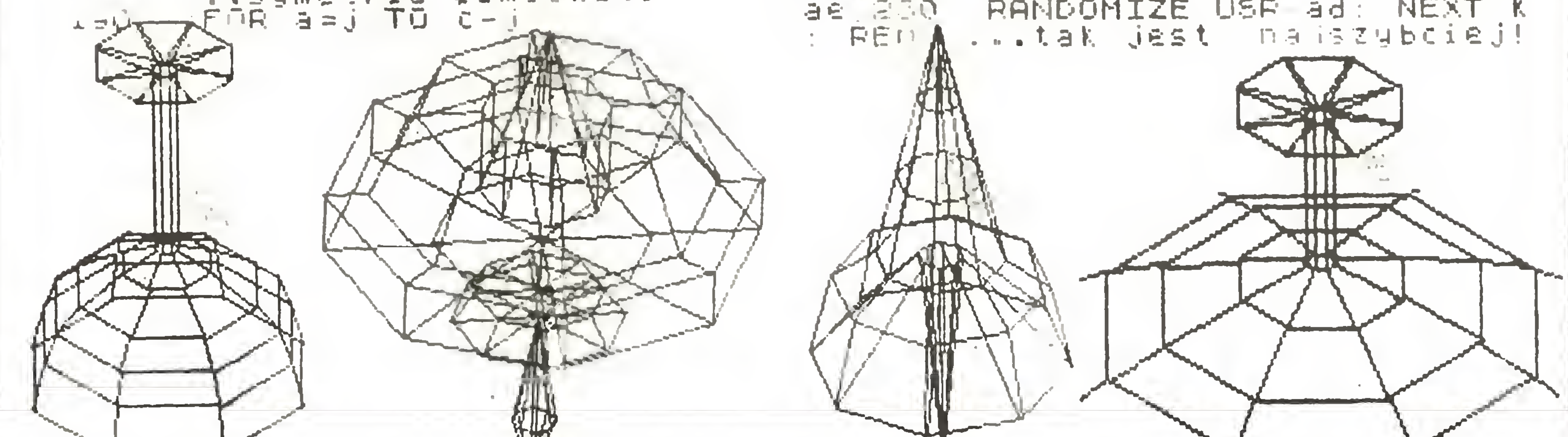
Dziwny sposób zapisu linii 450 wynika z potrzeby maksymalnego przyspieszenia jej działania. Zastąpienie w całym programie liczby 1 zmienną j skraca program o ok. 100 bajtów, co jest konieczne dla jego sprawnej pracy.

wg pomysłu z ZX User Club (Monachium, marzec 1984).

```
   1 REM @!?▖??? GO SUB VAL <>
 (12 dowolnych znakow)
   5 LET s4=SIN (PI/4): LET j=1:
     LET s2=SIN (PI/24):
     LET c2=COS (PI/24):
     DIM x(16,8): DIM y(16):
     DIM z(16,8): REM (pkt,rys)
  10 LET ad = 5 + PEEK 23635
          +256*PEEK 23636:
 REM ...adres 1 linii programu

  15 REM ...linie 15-50  mozna
        usunac po wpisaniu kodu
        do 1 linii programu
  20 FOR a=ad TO ad+11
  30  READ b: POKE a,b
  40 NEXT a
  50 DATA 33,  0, 64, 17,  0,230
      ,  1,  0, 24,237,176,201

  60 CLS : PRINT AT 0,10;
        "K A R U Z E L A"'
   "Podaj kolejne punkty przekr
oju  bryly. X>127 konczy zapis."
  70 FOR c=j TO 16
  80  INPUT "X=";x(c,1):
   IF x(c,1)>127 THEN GO TO 150
  90  INPUT ,"Y=";y(c):
      LET z(c,j)=0
 100  IF x(c,j)<0 OR y(c)<-30 OR
        x(c,j)>60+2*y(c)      OR
        x(c,j)>290-2*y(c)     OR
        y(c)>145 THEN GO TO 80
 120  IF c>j THEN
DRAW x(c,j)-x(c-j,j),y(c)-y(c-j)
 130  PLOT x(c,j)+128,y(c)+30
 140 NEXT c
 150> REM ...obracamy bryle i
           rysujemy 6 obrazkow
 160 FOR q=0 TO 5: CLS
 170  POKE ad+5,110+24*q: REM
   ...adres kolejnego obrazka
 180  FOR b=2 TO 8: REM
     ..symetria osmiokata
 190   FOR a=j TO c-j
 200     LET x(a,b)=s4*(x(a,b-j)-
                     z(a,b-j))
         LET z(a,b)=s4*(z(a,b-j)+
                     x(a,b-j))
 210   NEXT a: NEXT b
 240  FOR b=j TO 7
 250   FOR a=j TO c-2
 260    PLOT 128+x(a,b),
             30+y(a)-.5*z(a,b)
 270    DRAW x(a,b+j)-x(a,b),
        -.5*(z(a,b+j)-z(a,b))
 280    DRAW x(a+j,b+j)-x(a,b+j),
             y(a+j)      -y(a)
       -.5*(z(a+j,b+j)-z(a,b+j))
 290  NEXT a: NEXT b
 310  FOR a=j TO c-2
 320   PLOT 128+x(a,8),
            30+y(a)-.5*z(a,8)
 330   DRAW x(a,j)-x(a,8),
       -.5*(z(a,j)-z(a,8))
 340   DRAW x(a+j,j)-x(a,j),
            y(a+j)-y(a)-
        .5*(z(a+j,j)-z(a,j))
 350  NEXT a
 360  RANDOMIZE USR ad: REM
  >   ...zapamietaj obrazek
      ...nastepny nieco obrocony
 380 FOR a=j TO c-j:
  LET xn    =x(a,j)*c2-z(a,j)*s2
  LET z(a,j)=z(a,j)*c2+x(a,j)*s2
  LET x(a,j)=xn: NEXT a
 420 NEXT q: BEEP .5,30
 430 REM ...koniec pracy,czas
           na karuzele
 440 POKE ad,17: POKE ad+3,33:
     REM ...odczyt z pamieci
 445 LET ae=ad+5
 450 FOR k=0 TO j STEP 0:  POKE
ae,110: RANDOMIZE USR ad:  POKE
ae,134: RANDOMIZE USR ad:  POKE
ae,158: RANDOMIZE USR ad:  POKE
ae,182: RANDOMIZE USR ad:  POKE
ae,206: RANDOMIZE USR ad:  POKE
ae,230: RANDOMIZE USR ad: NEXT k
 : REM ...tak jest najszybciej!
```



SERIA I c.d.

Wiele gier komputerowych polega na odnalezieniu jednej, jedynej drogi wyjścia z labiryntu. Dla podniesienia atrakcyjności gry w poszukiwaniach wyjścia przeszkadzają nam jakieś duchy, potwory i zaczarowane przedmioty. Zwykle jednak sam labirynt nie zmienia się, a więc po kilku grach znamy go na pamięć.

Proponuję więc jako zadanie rozwiązać problem losowego generowania labiryntu, mającego dokładnie jedno wyjście (tzn. jedną drogę prowadzącą ze środka do wyjścia). Przez losowe generowanie rozumiem takie tworzenie labiryntu, aby nowy generowany labirynt był z dużym prawdopodobieństwem inny niż poprzednie.

Rozwiązaniem może być dokładny opis algorytmu takiego generowania. Z radością przyjmiemy także gotowe programy realizujące to zadanie w formie graficznej.

* * *

W teorii liczb (jednym z najstarszych działów matematyki) korzysta się często z tzw. funkcji Eulera. Funkcja ta, określona na liczbach naturalnych, przyjmuje jako swoje wartości liczbę mniejszych od badanej liczb pierwszych. Na przykład wartością funkcji Eulera dla liczby 9 będzie 4, bo mamy tylko cztery liczby pierwsze mniejsze od 9 (są to oczywiście 2, 3, 5, 7).

Proponuję napisać program, który mając jako daną wejściową pewną liczbę naturalną, podaje na wyjściu wartość funkcji Eulera obliczoną dla tej liczby.

* * *

Wielokrotnie, gdy tworzyłem jakieś obrazki na moim komputerze, stawałem przed problemem uzyskania lustrzanego odbicia tego co już narysowałem. Gdy rysunek był utworzony programowo, kłopotu dużego nie ma. Gorzej jest, jeżeli używałem do tego celu dyrektyw bezpośrednich. Podobny problem występuje w sytuacji, gdy chcemy "dodać" dwa utworzone wcześniej rysunki, to znaczy wyświetlić na ekranie rysunek przechowywany w pamięci nie niszcząc jednak tego co już zostało namalowane. Tego typu problemów można wymyślić wiele.

Proponuję napisać program, który będzie realizował operacje tego typu na uprzednio przygotowanych i przechowywanych w pamięci obrazkach. Punktować będziemy dwa aspekty tych programów: ilość i różnorodność możliwych operacji oraz funkcjonalność (tzn. łatwość wykorzystania wszystkich możliwości).

KINO I KOMPUTERY
WOJNY GWIEZDNE NA JAWIE [1]

Zawarty w 1972 r. układ ZSRR i USA w sprawie systemów obrony przed rakietami balistycznymi zakazywał dalszego rozwijania antyrakiet i systemów radarów wczesnego ostrzegania. Przed użyciem broni jądrowej miała obie strony powstrzymywać świadomość, że po odpaleniu rakiet kataklizm jądrowy jest nieuchronny dla nich obu.

Podjęcie przez USA prac nad "wojnami gwiezdnymi" Ronalda Reagana jest jawnym naruszeniem tego traktatu. Obok jednak aspektów i celów politycznych z "Inicjatywą Obrony Strategicznej" – tak brzmi oficjalna nazwa – wiążą się potężne interesy ekonomiczne, a także... fascynujące problemy o charakterze czysto inżynierskim łączące się z podstawami najszerzej rozumianej kultury ludzkiej.

Krótko o interesach: choć w USA prowadzi się obecnie śledztwo o nadużycia przeciw 45% wielkich dostawców sprzętu wojskowego, koncerny zbrojeniowe nadal mają w USA najwięcej do powiedzenia w sprawach wydatków militarnych. Dzięki "gwiezdnym wojnom" będą one latami dobrze żyły najpierw z badań, a następnie z prototypów i wreszcie z wytwarzania samego systemu.

* * *

Termin "wojny gwiezdne" wymyśliła amerykańska prasa. Film o tym tytule widzieli prawie wszyscy, a proklamowana przez szefa państwa inicjatywa była równie filmowa. Fantastyczny był też do niej komentarz: dzięki "wojnom gwiezdnym" broń jądrowa stanie się przestarzała (bo nieskuteczna), a jak już USA zainstalują w kosmosie swe urządzenia, to wszystko to udostępni się być może nawet ZSRR...

Istotą "gwiezdnych wojen" jest, w skrócie, niszczenie rakiet przeciwnika i ich głowic zaraz po starcie i w locie. Powstać ma wielopoziomowy parasol ochronny z najróżniejszych broni: dział elektronowych wyrzucających pociski z prędkością 25 km/s, dwudziestokrotnie większą niż w artylerii klasycznej i superlaserów rentgenowskich, dla których źródłem energii byłaby eksplozja jądrowa – fala uderzeniowa rozchodzi się wolniej niż promieniowanie i laser taki zdążyłby je zebrać, przetworzyć i wysłać w kosmos, gdzie ogromne zwierciadła odbiłyby je w stronę startujących rakiet. Rakiet może wystartować i parę tysięcy jednocześnie, lekarstwo więc (tysiące "kontrolowanych" eksplozji nuklearnych na własnym terytorium) byłoby równie gorzkie jak choroba, ale im też by się dostało...

Potrzebne będą też ogromne sieci rozmaitych czujników umożliwiających wykrycie zwiastunów startu i pozwalających na nakierowanie broni i kontrolę jej skuteczności, hale operacyjne przypominające ośrodki kontroli lotów kosmicznych, służby meteo, systemy ostrzegania o nadejściu np. deszczu meteorytów i wiele innych jednostek pomocniczych.

Wszystko to musiałoby być w pełni zautomatyzowane, gdyż na skuteczne zniszczenie celów przez system "wojen gwiezdnych" można liczyć tylko podczas ich startu – dopóki działają silniki. W tej fazie celów może być najwyżej kilka tysięcy. Gdy rozpoczną lot balistyczny, podzielą się one na dziesiątki tysięcy głowic i setki tysięcy atrap, bez oddania do nich "próbnego strzału" i przeprowadzenia skomplikowanej analizy danych o ich ruchu dla radarów i czujników nierozróżnialnych od prawdziwych głowic. Gdy głowice bojowe wchodzą w atmosferę z ponownie włączonymi silnikami, jest już zbyt późno. Nawet ich zniszczenie nie chroni już przed eksplozją i jej skutkami.

Wymogi wobec systemu

Na zniszczenie głowic pozostaje więc od 5 do 10 minut, a już teraz mówi się, że faza startu ulegnie skróceniu do 1...2 minut. Technicznie jest to możliwe, a powodów do zmiany silników nie brakuje. Jednym z nich jest oczywiście SDI.

Zatem w ciągu 1-10 minut cały, niezwykle skomplikowany system będzie musiał dokonać oceny zagrożenia oraz określić jakich środków należy użyć. Część z tych decyzji jest natury politycznej, część sztabowej. Tylko znikoma ich część to decyzje wojskowe typu operacyjnego, które może podejmować odpowiednio upoważniony oficer.

Są plany włączenia do "wojen gwiezdnych" także broni ofensywnych, czyli strategicznych sił jądrowych. W USA siły morskie, lotnicze i lądowe odznaczają się dużą autonomią i mają własne rakiety zdolne obrócić świat w perzynę. Owo włączenie polegałoby na tym, że przy przekroczeniu pewnego progu liczby celów, system komputerowy sterujący "wojnami gwiezdnymi" podejmowałby decyzję o odpaleniu w odwecie odpowiedniej liczby rakiet strategicznych. Zatem "wojny gwiezdne" to nie tylko czysta obrona przed "imperium zła" (jak prezydent Reagan nazwał ZSRR), ale i równoczesny atak. Czy jednak wszystko odbyłoby się z iście komputerową precyzją? Należy wątpić.

Czy istnieje system podobny do "wojen gwiezdnych"

Amerykanie dysponują czymś porównywalnym z "wojnami gwiezdnymi". Jest to World-wide Military Command and Control System w siłach lotniczych. Obok prezentujemy schemat ideowy, wprawdzie bardzo ogólny, ale za to wyjęty z dokumentów Pentagonu, więc wierny co do zarysów (chyba). WWMCCS powstał w 1977 r. kosztem od 10 do 15 mld dolarów. Obsługuje go 10 komputerów o mocy obliczeniowej 1,5 mln operacji zmiennoprzecinkowych na sekundę. Program liczy sobie 10 mln linii a przecież nie ma tu tak rygorystycznych wymogów czasowych. WWMCCS – istniejący już od prawie dziesięciu lat – przez 62 proc. czasu pracy jest w naprawach, bowiem coś się w nim ciągle psuje. Gdy działa, powstają sytuacje niejasne, których programiści nie przewidzieli. Nie zawsze jest tak, jak przy pierwszych lotach w kosmos, gdy pojazd Gemini 5 spadł na Ziemię 160 km od przewidywanego punktu, bowiem programiści zapomnieli uwzględnić ruch naszej planety po orbicie... Bywają sytuacje nie poddające się ocenie nawet najlepiej zaprogramowanych komputerów. Gdy system wykrywa zagrożenie – ogłasza alarm; następnym etapem jest zwołanie tzw. konferencji oceny zagrożenia. Wypadki podnoszenia fałszywego alarmu były bardzo liczne. Według niektórych wojskowych amerykańskich, gazety mogłyby prowadzić stałą rubrykę z takimi "wydarzeniami". Przypadki zwołania konferencji oceny zagrożenia były rzadsze, ale za to znamienne. Np. 15 marca 1980 r. start ćwiczebnej rakiety średniego zasięgu na terytorium radzieckim poczytano za atak jądrowy.

Liczenie na to, że można zautomatyzować ocenę zagrożeń militarnych, jest więc błędne. Znawcy nauk politycznych przypominają, że I wojna światowa wybuchła wprawdzie głównie wskutek niekłamanej chęci zainteresowanych rządów, ale na ostateczną decyzję bezpośredni wpływ wywarły informacje napływające z wywiadów, a konkretnie od agentów. Ci zaś byli wprowadzeni w błąd przez kontrwywiady, które starały się przestraszyć agentów wroga tak, aby ich mocodawcom odechciało się atakowania... Skutek wszyscy znamy.

Nawet najlepszy komputer jest jak dziecko prostoduszny i właściwie bezbronny wobec prowokacji czy celowego wprowadzenia w błąd. Ten od "wojen gwiezdnych" miałby jednak zarazem podejmować decyzje o ataku jądrowym. Czasu – na obudzenie szefa sztabu i prezydenta, by ten mógł zdecydować, że lepiej sięgnąć po słuchawkę "gorącej linii" i zapytać wprost, co druga strona wyrabia, w myśl założeń "wojen gwiezdnych" – by nie było.

Jak oprogramować "wojny gwiezdne"

Niedawno prasa doniosła o wycofaniu pieniędzy rządowych z programu DIVAD. Miała to być podwójnie sprzężona armata przeciwlotnicza na samobieżnym podwoziu, której radar i komputer po wykryciu celu w każdych wa-

runkach i na każdej wysokości wydawałby polecenie automatycznego otworzenia ognia i zniszczenia samolotu wroga. Ale po niemal dziesięciu latach pracy i wydaniu ponad dwu miliardów dolarów – system DIVAD nie chciał zestrzeliwać samolotów. Niemal zawsze chybiał, choć raz był bardzo blisko rozwiązania problemu, czyli spotkania pocisku z celem. Podczas strzelania pokazowego wobec licznie zebranych sztabowców i ekspertów, armata chyżo skierowała lufy w stronę celu, ale go minęła, następnie wykonała obrót i skierowała się w stronę trybun honorowych... Oficerowie się rozbiegli, a gdy już wrócili, wycofali swe poparcie finansowe. A przecież oprogramowanie komputera, który miałby sterować takimi armatami, jest niezwykle proste w porównaniu z systemem "wojen gwiezdnych"... Specjaliści byli zdania, że DIVAD się skompromitował, gdyż nie przewidział wszystkiego, każdych warunków. Racja, ale problem z "wojnami gwiezdnymi" pozostaje.

Program zestrzeliwania samolotów można przetestować na poligonie w warunkach identycznych z sytuacją na polu walki. Gorzej ze strzelaniem w kosmosie, niezależnie od tego, czy chodziłoby o lasery czy o działa elektromagnetyczne. Nie można zaś wyobrazić sobie mozolnego testowania programu sterowania owymi zwierciadłami wiszącymi w kosmosie, które miałyby odbijać twarde promieniowanie powstałe w wyniku eksplozji jądrowej na Ziemi...

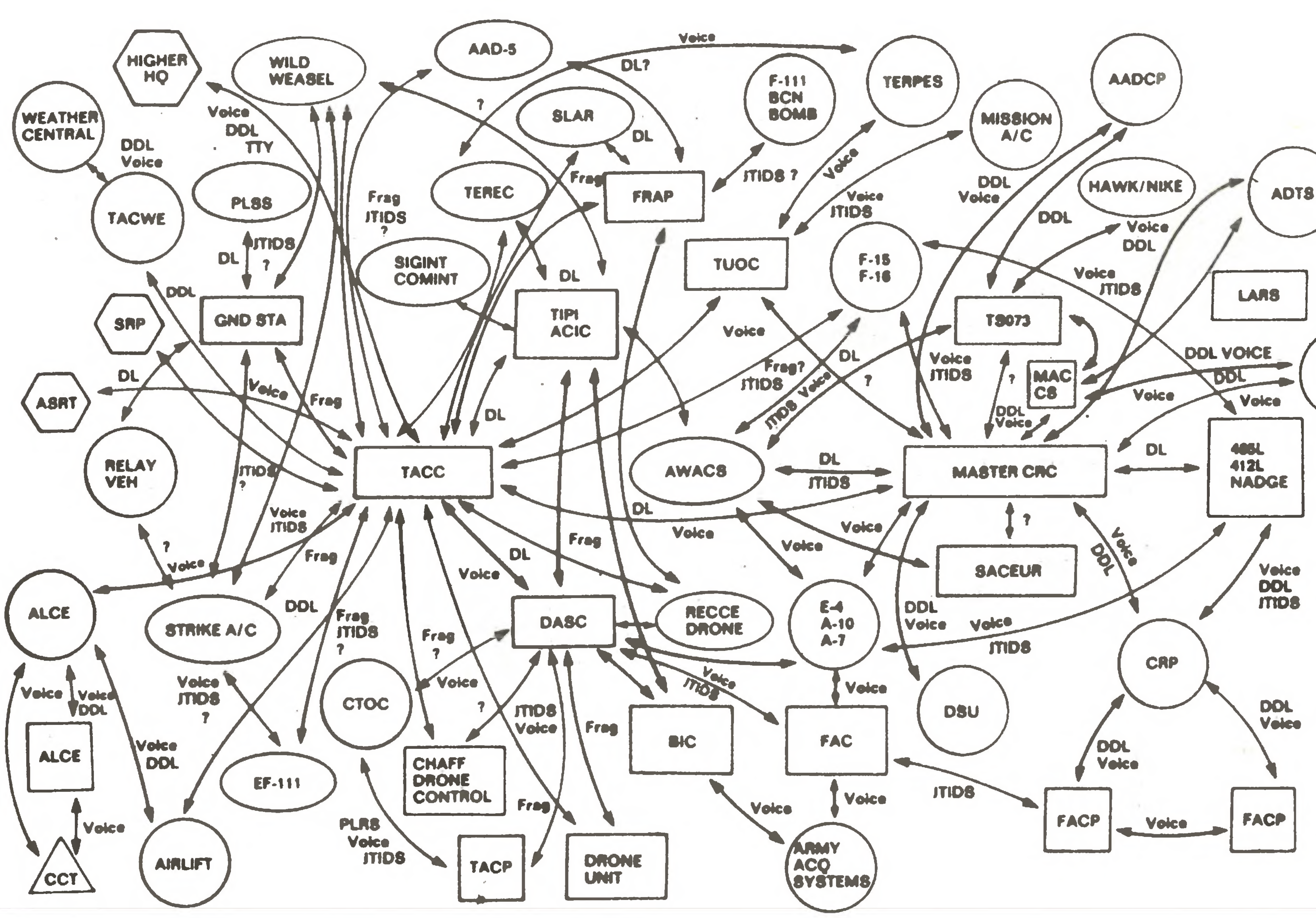
Jedno z praw Murphy'ego (patrz na str. 40) głosi, że "jeśli coś może pójść źle – na pewno tak się stanie", a inne, że "nie ma pracy tak prostej, aby nie można było jej źle wykonać". System DIVAD – aby odwołać się do ostatniego przykładu – wykazał, że oba prawa działają. Na jakiej zatem podstawie przypuszcza się, że w "wojnach gwiezdnych" wszystko będzie przebiegać zgodnie z założeniami? Czy to chęć przeżycia (lub też nieprzeżycia?) kina na jawie?

Poza mnogością elementów systemu i niemożnością starannego testowania go w warunkach choćby zbliżonych do naturalnych są jeszcze inne poważne problemy. Załóżmy, że cały system zostanie zbudowany, a odpowiednią aparaturę Amerykanie umieszczą w kosmosie. Może się wtedy okazać, że założenia strategiczne, według których automaty miały by prowadzić wojny, są błędne. Przypomnę dwa przykłady błędnych doktryn wojskowych. Pod wpływem doświadczeń I wojny światowej, Francja zbudowała w latach 1929-34 linię Maginota, podobno nie do zdobycia. Wszakże Niemcy, ożywiani zupełnie inną doktryną blitzkriegu, w 1940 r. po prostu obeszli ją z lewej flanki. Drugi przykład – to mniej znana doktryna włoskiego generała Dohoueta przekonanego, że przyszłą wojnę rozstrzygnie się przez bombardowanie strategiczne. Potężne armady bombowców miały siać zniszczenie i łamać ducha walki na głębokich tyłach, tak jak na froncie. Postępując zgodnie z tym przekonaniem Wielka Brytania i USA prowadziły przez dużą część II wojny światowej dywanowe naloty na ośrodki niemieckiego przemysłu zbrojeniowego. Szczególną uwagę poświęcono fabrykom samolotów. Tymczasem produkcja samolotów w nazistowskich Niemczech była największa pod koniec 1944 r., już po owych strategicznych nalotach, a zaczęła się zmniejszać dopiero pod wpływem braku materiałów. O innych doktrynalnych błędach powodujących klęski na polach bitew już nie będę wspominać.

Powstaje pytanie, czy można stworzyć oprogramowanie mądrzejsze od jego twórców? Czy automatyczne prowadzenie bitew ze startującymi i lecącymi rakietami będzie mogło odbywać się według określonego planu? Czy przeciwnik nie domyśli się, jaki będzie ten plan? Jak uchronić systemy komputerowe przed celowymi zakłóceniami i próbami wprowadzenia w błąd? Odpowiedź może być tylko jedna – programiści ustawią cały system tak, by na wszelkie zagrożenia odzew był natychmiastowy. Tak właśnie działa amerykańska policja: "Najpierw strzelaj, pytaj potem". Istnieją więc co najmniej dwie możliwości. Pierwsza – że zbyt złożony system może zacząć funkcjonować pod wpływem nieprzewidzianych czynników i nie jest to sytuacja wydumana, mamy tego przykłady. Druga – to załamanie się systemu pod wpływem warunków bitwy o kosmos. Na szczęście w tym przypadku brak jeszcze doświadczeń, ale awarie komputerów bojowych pod wpływem ich przeciążenia można już sobie wyobrazić.

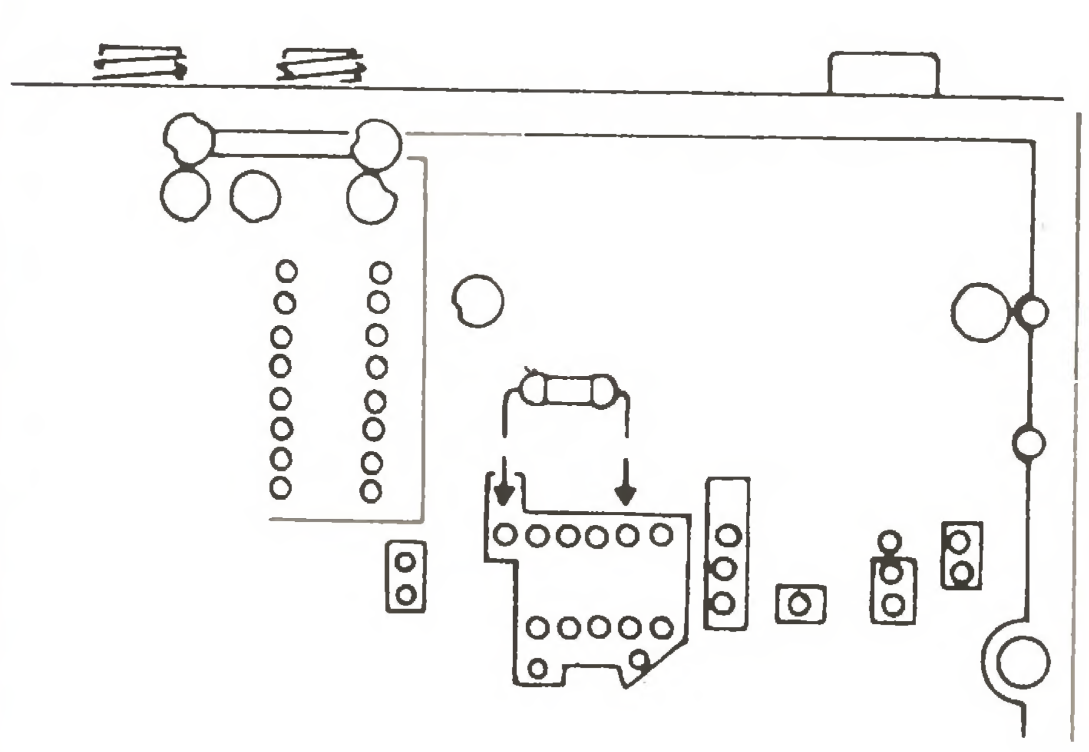
W jednym z kolejnych numerów "Komputera" – o problemach oprogramowania "gwiezdnych wojen".

JACEK A. LIKOWSKI



STOP w Spectrum Plus

Jedną z najczęstszych awarii, a właściwie usterek ujawniających się w ZX Spectrum Plus jest niewłaściwe funkcjonowanie przycisku STOP (SYMBOL SHIFT +A). Usterka ta jest szczególnie irytująca, gdyż pojawia się i znika w najbardziej nieoczekiwanych momentach, a poza tym przejawia się w bardzo nietypowy sposób: oba klawisze z osobna działają poprawnie, cała klawiatura spisuje się pozornie bez zarzutu, a ta jedna kombinacja klawiszy zawodzi. Wielu użytkowników próbuje po rozkręceniu komputera znaleźć przyczynę w braku kontaktu lub zwarciu wewnątrz klawiatury, co nie daje żadnego rezultatu, natomiast często kończy się uszkodzeniem delikatnej folii klawiatury.



Tymczasem przyczyna tego dziwnego zachowania się komputera (objaw ten praktycznie nigdy nie występuje w zwykłym ZX Spectrum) leży – o dziwo – poza klawiaturą. Okazuje się, że zmienione połączenia wewnątrz klawiatury powodują zwiększone obciążenia części obwodów i w rezultacie stany elektryczne nie ustalają się dostatecznie szybko.

Radykalnym rozwiązaniem zalecanym przez producenta w instrukcji fabrycznej dla osób samodzielnie dokonujących przeróbki ZX Spectrum na ZX Spectrum Plus ("ZX Spectrum +Upgrade Instructions) jest dolutowanie w miejscu oznaczonym na płytce komputera symbolem R68 (we wcześniejszych modelach ZX Spectrum jest to tylko oznaczone miejsce, bez żadnego wlutowanego opornika) – opornika o oporze ok. 22 kiloomów (w praktyce wartość oporu może różnić się do ok. 25% od podanej).

W najstarszej wersji ZX Spectrum (Issue 1) nie było na płytce oznaczenia R68. Posiadacze tej wersji muszą wlutować opornik na odwrotnej stronie płytki w miejscu zaznaczonym na rysunku.

(WM)

Dziękujemy!

Prawdę mówiąc nie spodziewaliśmy się aż tak dużego oddźwięku po pierwszym numerze „Komputera" – zwłaszcza że konkurencja nie śpi. Przeżywamy nieskrywaną radość, że od początku mamy tak szerokie grono czytelników. Będziemy starali się drukować listy najciekawsze i te, których tematyka najczęściej się będzie powtarzać.

* * *

Z niecierpliwością czekałem na ukazanie się pierwszego numeru Waszego miesięcznika. Moje ciche nadzieje w całości się spełniły, jest to pismo poważnie traktujące problem edukacji komputerowej naszego społeczeństwa.

mgr inż. Jerzy Szczerbiński
Zielona Góra

„Popularny Miesięcznik Informacyjny" to niezaprzeczalne osiągnięcie Waszego zespołu. Wypełnia on pewien obszar „czarnej plamy" jasnym światłem postępu w dziedzinie wyjątkowo zaniedbanej. Osobiście szczególnie interesuje mnie temat edukacji w szkołach podstawowych.

Aleksander Jabłoński
Gliwice

Wasze pismo jest dobrym pismem o mikrokomputerach. Bardzo chciałbym, aby w tym piśmie były drukowane programy gier i dydaktyczne. W „Komputerze" można by było drukować plany urządzeń dodatkowych oraz wysyłkowo sprzedawać kasety z programami.

Piotr Chałupka
Gorlice

Nie uniknęliśmy tradycyjnego chochlika drukarskiego, który dał o sobie znać i to od razu na okładce. Oczywiście jesteśmy miesięcznikiem INFORMATYCZNYM, a nie jak to mylnie wydrukowano, informacyjnym. Edukacja komputerowa jest jednym z celów naszej działalności, tak więc będziemy się starali zaspokajać zapotrzebowanie na tę tematykę. W naszych planach leży także seryjna produkcja kaset magnetofonowych z programami, których ceny jak już wspominaliśmy będą konkurencyjne w stosunku do oferowanych na „perskim". Oczekujemy w związku z tym na sygnały od autorów oryginalnych, powstałych w Polsce, programów. Pragniemy także otworzyć „sklep fabryczny", ale chwilowo to przyszłość.

* * *

Proszę o podanie adresów firm, które prowadzą wysyłkową sprzedaż komputerów (RFN, Anglia).

Zbigniew Szwajlik
Jastrzębie

Ten i podobne listy stanowią sporą część nadsyłanej do nas korespondencji. Niestety nie możemy podawać adresów zagranicznych firm prowadzących wysyłkową sprzedaż komputerów osobistych, ponieważ po pierwsze nie mamy na to ich zgody, a po drugie nie moglibyśmy przyjąć odpowiedzialności za sposób wywiązywania się ze zleceń. Z osobistych doświadczeń wiemy, że bardzo różnie różne firmy reagują na chęć kupna wyrażoną na odległość. Nie wszystkie w ogóle podejmują korespondencję, niektóre przeciągają sprawy w czasie, ponad cierpliwość entuzjastów mikrokomputeryzacji. Zainteresowanym polecamy uważną lekturę ogłoszeń, także w naszym piśmie.

* * *

ORTOGRAFIA – pierwszy godny recenzji polski program edukacyjny. Taki tytuł z pewnością zachęca do zapoznania się z tą recenzją. A jakie wnioski wyciąga autor? Po pierwsze stwierdza ponad wszelką wątpliwość, że główną zaletą ORTOGRAFII jest sam fakt jej istnienia. A zaraz potem wylewa kubeł zimnej wody na głowę Jacka Potempy i doszukuje się wszelkiego złego co tylko w ORTOGRAFII można znaleźć.

Zdaniem Pana Marka Cara program jest mało atrakcyjny, naciska się tylko dwa klawisze, nie wiadomo dlaczego „zasuwka" nie pisze się „zasówka". Słowem program jest zły.

Czy na pierwsze wejście „Komputera" należało umieszczać taką recenzję?

Janusz Gajewicz
Wodzisław

Okazuje się, że wpajana intensywnie od lat zasada „pisać dobrze albo wcale" na dobre zagościła w świadomości czytelników. Nie miejsce tu na tłumaczenie, dlaczego uważam ją za błędną, wręcz szkodliwą. Podobnie jak dopisywanie ocen („słowem program jest zły"). W listopadzie ub. r. napisałem, co napisałem i większość krytycznych uwag, głównie dydaktycznych, odnosi się również do drugiej wersji „ORTOGRAFII". Umożliwienie rywalizacji dwóm osobom oraz wprowadzenie punktacji przydało grze widowiskowości i uczyniło ją bardziej atrakcyjną, choć nadal bardziej tresuje ona niż uczy, co jest wszak głównym zadaniem programu edukacyjnego. Tym niemniej zdecydowani jesteśmy, po dokonaniu przez autora niezbędnych poprawek, zaproponować mu wydanie tego programu w naszym wydawnictwie kasetowym. I to właśnie jest miarą oceny „ORTOGRAFII". Mimo zastrzeżeń.

MAREK CAR

BŁĘDY W SPECTRUM

Zgodnie z wcześniejszymi obietnicami rozpoczynamy wydawanie serii specjalnych zeszytów ABC "Komputera". W pierwszym Andrzej Kadlof zebrał wszystkie informacje niezbędne każdemu użytkownikowi ZX Spectrum – zarówno zawarte w instrukcji firmowej, jak i pominięte w niej.

Publikujemy dziś fragment tej broszurki. Znajdziecie w niej także opis podstawowych procedur zawartych w ROM Spectrum, opis działania kalkulatora ZX Spectrum, sposoby wykorzystywania przez ZX Spectrum kanałów wejścia/wyjścia, dokładne wyjaśnienie funkcji zmiennych systemowych, organizację obsługi klawiatury – wszystko, co potrzebne, by praca na Spectrum była prawdziwą radością.

Kolejne tomiki poświęcone będą Commodore 64, LOGO, językowi Pascal, MS-Basic – dialektowi Basicu stosowanemu w IBM PC. Planujemy także poradnik dla tych, którzy pragną sami zbudować własne Spectrum.

ABC "Komputera" – już wkrótce w kioskach!

BŁĘDY W SPECTRUM

Autorzy systemu operacyjnego i interpretera BASICA w ZX SPECTRUM wykonali wspaniałą robotę. Nie zdołali jednak ustrzec się przed kilkoma błędami. Niektóre z nich są mało istotne, inne zaś nieświadomego programistę mogą doprowadzić do ciężkiej frustracji. Poniżej przedstawiamy listę znanych nam błędów ku przestrodze wszystkim użytkownikom tego świetnego mikrokomputera.

1. Błąd dzielenia. Pod adresem 3200 umieszczono wartość E1 zamiast DA. W konsekwencji gubiony bywa ostatni bit, co prowadzi do niewłaściwych zaokrągleń. Skutki błędu demonstruje poniższy program:

```
10 LET a=a/b
20 IF a THEN GO TO 10
30 PRINT "Osiągnięto zero"
```

Uruchamiając go z wartościami a=1 i b=3 po sekundzie uzyskamy wydruk "Osiągnięto zero". Uruchamiając go jednak ponownie, tym razem z wartościami a=1 i b=2 wpadamy w pętlę nieskończoną, bo SPECTRUM sądzi, że 2 ^ - 128 = 2 ^ - 128/2

2."-65536" błąd. Autorzy dopuścili się niekonsekwencji w przedstawieniu tej liczby. Raz jest ona przechowywana w postaci zmiennoprzecinkowej a innym razem jako liczba całkowita w kodzie uzupełnień do dwóch. Skutki niejednoznaczności objawiają się na przykład w następującym rozkazie PRINT INT -65536. Na ekranie pojawia się liczba.... - 1.

3. CHR$ 8 błąd. Wydruk tego symbolu kontrolnego powinien przesunąć kursor na ekranie o jedną pozycję w lewo lub na koniec poprzedniej linii. Tak istotnie jest w liniach 1...23. Na skutek błędu nie można przejść z początku pierwszej linii na koniec tej o numerze 0. Do "ciekawych" efektów prowadzi próba przesuwania kursora w lewo od pola (0,0). Radzimy to sprawdzić, ale raczej bez ważnych danych w pamięci komputera.

4. CHR$ 9 błąd. Ten symbol powinien dla odmiany przesuwać kursor o jedno pole w prawo. Tu popełniono jednak poważniejszy błąd. W rezultacie jego wydruk praktycznie nie robi nic. Wykonane są bowiem wszystkie niezbędne obliczenia, ale zapomniano zmodyfikować zmienne systemowe.

5."Press any key..." błąd. Co najmniej w kilku sytuacjach SPECTRUM przerywa pracę i czeka na ponaglenie użytkownika przez wciśnięcie któregokolwiek klawisza. Błąd sprowadza się do tego, że komputer nie reaguje na klawisz CS ani SS choć reaguje na oba razem.

6. Błąd wskaźnika linii aktualnej. Przypuśćmy, że ostatnia linia w programie ma np. numer 1000. Po wciśnięciu 1001 i ENTER a następnie CS/1 na dół ekranu zostanie skopiowana linia 1000, ale razem ze wskaźnikiem linii aktualnej, który przed odesłaniem linii trzeba osobno kasować.

7. Błąd DELETE. Przy kasowaniu zawartości dolnego ekranu przez CS/1 zostaje na dół skopiowana linia aktualna programu i nie jest przywracany właściwy rozmiar tego obszaru. W praktyce trzeba ponownie wciskać ENTER.

8. Błąd wiodących spacji. Niektóre słowa kluczowe są w czasie wydruku poprzedzane spacją. Okazuje się jednak, że nie zawsze. Spróbujcie np. wykonać PRINT CHR$ 255 ; CHR$ 13 ; CHR$ 255.

9. Błąd trybu K. Po wciśnięciu klawisza w trybie K i jego przytrzymaniu klawisz zaczyna się powielać. Kursor się zmienia na L lub C, ale drukowany jest cały czas symbol w trybie K.

10. SCREEN$ błąd. Drobny błąd pod adresem 257D (C3 zamiast C9) objawia się w sytuacjach takich jak w poniższym przykładzie

```
10 PRINT "1234567890"
20 LET a$=SCREEN$ (0,0) + SCREEN$ (0,1)
30 PRINT a$
```

Zamiast spodziewanego 12 na ekranie pojawia się 22. Jeszcze dziwniejsze wydruki otrzyma się, gdy w linii 20 dopisać na końcu jeszcze + SCREEN$ (0,2) + SCREEN$ (0,4). (Zmienna a$ powinna przyjąć wartość "55").

Ten błąd łatwo ominąć dodając do a$ wartości SCREEN$ (0,1) pojedynczo, a nie w jednym wyrażeniu.

11. STR$ błąd. Działając z liczbami -1 <x< 1, ale różnymi od zera można się naciąć na niezwykłą "żarłoczność" rozkazu STR$. Spróbujcie poniższych rozkazów

```
PRINT "ALA"+"Bum cyk cyk" + STR$ .001
PRINT 7 + VAL STR$ .001
```

W obu wypadkach Waszym zdumionym oczom na ekranie ukaże się jedynie .001. Winę ponosi SPECTRUM!

12. CLOSE błąd. Próby odłączenia strumienia 4... 15 od kanału przed uprzednim jego przyłączeniem prowadzą do nieprzewidzianych efektów z restartem systemu włącznie. A wszystko dlatego, że w tablicy zawierającej dane o kanałach w ROM-ie pod adresem 1716 zapomniano umieścić znacznik końca tablicy.

13. "RET" błąd. Jest to jeden z trudniejszych do zlokalizowania błędów. Objawia się czasami przy powrocie do BASICA z programu w kodzie maszynowym uruchamianym przez USR k. Zapomniano bowiem przed powrotem do interpretera odtworzyć wartość pary rejestrów H' L'. Jeśli Wasz program ich używał i zmodyfikował, to co się stanie przy powrocie do BASICA, jest kwestią przypadku. Zazwyczaj jest to zawieszenie się systemu, ale nie zawsze. Co najciekawsze wartość ta jest zawsze stała i powinna wynosić 2578_{HEX} = 10072.

14. Błąd NMI. Jest to jeden z najpoważniejszych błędów. Przestawienie jednego bitu pod adresem 006D sprawiło, że SPECTRUM nie jest w stanie przyjmować tak zwanych przerwań niemaskowalnych z obsługą przerwania przez procedurę użytkownika. Przerwania takie mogą być bądź ignorowane bądź restartować system przez skok do adresu 0. W szczególności uniemożliwia to kontrolowany restart systemu po jego załamaniu jak i cały szereg innych zastosowań komputera. Eliminacja tego błędu wymaga niełatwych i kosztownych przeróbek technicznych.

15. PAUSE n błąd. Błąd w procedurze obsługującej samopowtarzalność klawiszy sprawia, że jeśli jakiś klawisz był naciskany tuż przed wykonaniem instrukcji PAUSE, to jest ona ignorowana. Zilustrujmy to przykładem.

```
10 PRINT "Puść klawisz, gdy usłyszysz dźwięk"
20 FOR i=1 TO 500:NEXT i
30 BEPP 1, 10
40 PAUSE 0
50 PRINT "Koniec"
```

Zachowanie się programu niczym nie wskazuje na istnienie w nim linii 30.

16. CLS błąd. Przy wartościach zmiennej systemowej DF SZ mniejszych od 2 trudno uznać efekt tego rozkazu za czyszczenie ekranu. Dla DF SZ = 1 można to uznać za ciekawostkę ale przy DF SZ = 0 to już tragedia! Sprawdźcie!

Mamy nadzieję, że powyższa lista w pełni wyczerpuje "niespodzianki", na które można się natknąć programując ZX SPECTRUM.



GIEŁDA

COMMODORE 128 D

Nie ma niezawodnej recepty na sukces i powodzenie nowego modelu komputera – nikt tego nie wie lepiej niż Commodore, który próbował powtórzyć sukces C64 modelami C16 i Plus-4. Oba zalegały półki, jeśli nie liczyć pewnych zrywów sprzedaży z okazji gwiazdkowej obniżki cen. Jednocześnie stary C64 sprzedawał się przez cały rok 1985 w znacznych ilościach. Było jednak dla Commodore oczywiste, że czteroletnie już C64 potrzebuje następcy. Jedynym rozsądnym i oczywistym rozwiązaniem było zbudowanie nowego modelu całkowicie kompatybilnego z C64, a jednocześnie znacznie od niego lepszego. I tak w 1985 roku narodził się Commodore 128 z wbudowanym procesorem Z80, systemem CP/M, językiem BASIC 7.0, 128 KB RAM i trybem naśladującym C64. Od pierwszych chwil pojawienia się C128 Commodore zapowiadał wypuszczenie C128 z wbudowanym dyskiem. Zapowiedzi te traktowano jednak jako kolejne niespełnione marzenia – czyż nie zapowiadano przenośnego modelu z wyświetlaczem ciekłokrystalicznym (Commodore LCD) lub nie prezentowanego nawet (w Hanowerze) modelu kompatybilnego z IBM PC?

Tym razem jednak zapowiedzi potwierdziły się w pełni – Commodore 128D wszedł na rynek oferując wbudowany dwustronny disk drive 5.25 cala, dołączalną klawiaturę i monochromatyczny monitor za jedyne 499 funtów Plus VAT (w kwietniu cena spadła do 450 funtów).

Commodore 128D stał się więc realną alternatywą dla Amstrada PCW 8256 – procesora tekstu z systemem CP/M. Do C128D trzeba, co prawda, dokupić jeszcze drukarkę (za około 150 funtów), i edytor tekstu (za około 50 funtów), ale Amstrad nie gwarantuje programowej kompatybilności z bardzo popularnym C64 oraz ma gorsze możliwości graficzne i dźwiękowe. Przeciwnicy Commodore mogą jednak twierdzić, że za cenę systemu C128 (około 700 funtów) można mieć Amstrada PCW i Commodore 64.

C128D wygląda jak profesjonalny system biurowy. Klawiatura ma taki sam układ jak podstawowy C128, a odłączalność klawiatury pozwala na swobodne nią manipulacje bez obawy o odłączenie zasilania lub monitora (co mogło wystąpić w C128).

Zielony monitor 1901 ma bardzo wygodnie umieszczone z przodu wszystkie pokrętła regulacyjne, jest przyjemny dla oka i może tylko zbyt duży, ale za to bardzo czytelny.

Tak jak i wersja podstawowa, C128D oferuje dwie wersje języka BASIC-BASIC 7.0 i znany z C64 BASIC 2.0. Nowy BASIC bardzo dobrze wykorzystuje możliwości sprzętu. Komenda FAST pozwala korzystać (w trybie 80 kolumnowym) z pełnej szybkości oferowanej przez zegar 2MHz. Komendy CATALOG, BSAVE i BLOAD umożliwiają łatwą współpracę z dyskami, a BOOT pozwala na bezpośredni dostęp do pliku na dysku. Operacje dyskowe stają się jeszcze prostsze przy pracy pod CP/M, który w połączeniu z dyskami 5.25 cala pozwala czytać pliki zapisane w formacie MFM – przez Kaypro, Osborn One i IBM PC. Nie oznacza to kompatybilności z IBM PC, a tylko możliwość przenoszenia niektórych plików danych (zapisanych pod CP/M).

Stacje dysków typu 1571 są dwustronną wersją typu 1570 wypuszczonego w zeszłym roku dla podstawowego modelu C128. Stacja ma wbudowany procesor 6502, 32KB ROM i 2KB RAM co pozwala na automatyczne rozpoznawanie systemu pod jakim zapisana jest dyskietka (CP/M, BASIC 7.0 lub C64). Dane transmitowane są z prędkością 300 bodów dla C64 (stacje udają w tym trybie bardzo wolne stacje 1541) i 5200 bodów dla plików zapisanych pod CP/M.

Oprogramowanie jest chyba tym, co najsilniej przemawia na korzyść C128D (w porównaniu z Amstradem PCW). Większość z bogatej biblioteki gier i programów użytkowych Commodore 64 powinna dać się uruchomić na C128D bez żadnych modyfikacji, podobnie programy użytkowe systemu CP/M można przenieść na C128D bez większych zmian. Programy napisane specjalnie na podstawową wersję C128 pozwalają w największym stopniu wykorzystać możliwości sprzętowe – szczególnie szybkość w opcji 2MHz. Zapowiadane jest wprowadzenie na rynek nowych programów opracowanych specjalnie dla C128D – procesora tekstu Script 128 i zintegrowanego pakietu obliczeniowego Micro Clerk.

C128D sprawia wrażenie solidnego i dobrze zaprojektowanego sprzętu składającego się z trzech pudełek – monitora, CPU ze stacją dysków (stanowiącego podstawę dla monitora) oraz klawiatury. Pozwala to uniknąć plątaniny kabli oraz zajmuje mało miejsca. Dokumentacja systemu składa się z dwóch części. Pierwszej poświęconej C128 i zawierającej objaśnienia komend, przykłady programów itp. – podobnej do znanej z modelu C64, i drugiej – poświęconej współpracy z dyskami.

Commodore 128D wchodzi na rynek, by konkurować w klasie domowo-biurowych komputerów w cenie do 500 funtów z Amstradem i Atari. C128D wydaje się jednak trudny do pobicia ze względu na swą uniwersalność i solidną bazę programową. Można chyba śmiało stwierdzić, że Commodore 64 doczekał się w końcu godnego następcy.

Na podstawie Your Computer nr 3, 1986
opracował ZBIGNIEW BLEWOŃSKI

GIEŁDA W ABC

Dziś zamiast tradycyjnych informacji z perskiego jarmarku, proponujemy ofertę handlu państwowego: warszawski supersam z artykułami spożywczymi (!) ABC przy ul. Karolkowej rozszerzył swą ofertę o komputery. Na klientów pierwszego dnia sprzedaży czekały m.in.:

Spectrum Plus	210 tys. zł
Commodore 128 (sama klawiatura)	680 tys. zł
Amstrad 6128 z zielonym monitorem	1.000.000
(słownie złotych milion, groszy zero)	
z monitorem barwnym	1.150 tys. zł
Amstrad 8256	1.890 tys. zł
dyskietka 3"DS/DD (dwustronna, podwójna gęstość)	3680 zł

Zainteresowanie klientów umiarkowane, gospodyń domowych kupujących komputerek do bułki z masłem nie zauważyliśmy, stoisko jednak nie zbankrutuje: już w ciągu pierwszych dni sprzedaży zniknęły wszystkie ZX Spectrum.

Okazuje się, że dla wielu instytucji sklep państwowy, nawet drogi, ale wystawiający rachunki i umożliwiający sprzedaż od ręki, jest najlepszym miejscem zakupu – nawet jeśli oferowane warunki serwisu gwarancyjnego nie są jeszcze sprawdzone w praktyce.

* * *

Sygnalizowany przez nas spadek cen układów wysokiej skali integracji w krajach je produkującycych znajduje swój oddźwięk także na giełdzie. Szczegółowo ceny przedstawimy za miesiąc, dziś jeden przykład: Eprom 2764 można obecnie kupić za ok. 1000–1500 zł – cena wydaje się konkurencyjna nawet w stosunku do cen detalicznych w II obszarze płatniczym.

* * *

Giełda u sąsiadów: z ogłoszeń w Amatérské Radio nr 2/86

Počítač (czyli po czesku komputer) Commodore 116	10 700 Kčs
ZX Spectrum 48 K	9 500 Kčs
ZX 81 + 16 K RAM	5 500 Kčs
ZX Spectrum Plus, nový	9 000 Kčs
Mikropočítač Commodore 64 + datasette + joystick + padles	17 800 Kčs
Sharp PC-1211	5 500 Kčs

Podajemy najniższe oferowane ceny. Proporcje wydawać mogą się zaskakujące, ale nie wiemy, za ile faktycznie można sprzedać te urządzenia. Sądząc z liczby ofert u Czechów Sharp 1211 dorównuje popularnością ZX Spectrum i ZX 81, oferty na Commodore są jednostkowe.